

**Санкт-Петербургский филиал федерального государственного
автономного образовательного учреждения высшего образования
«Национальный исследовательский университет
"Высшая школа экономики"»**

Факультет Санкт-Петербургская школа
физико-математических и компьютерных наук
Департамент информатики

**Рабочая программа дисциплины
Алгоритмы и структуры данных**

для образовательной программы «Прикладная математика и информатика»
направления подготовки 01.03.02 «Прикладная математика и информатика»
уровень бакалавриат

Разработчики:


Близнец Иван Анатольевич, ibliznecz@hse.ru

Мишунин Александр Сергеевич, amishunin@hse.ru

Утверждена Академическим руководителем образовательной программы

«31» августа 2018 г.

А.В.Омельченко



Санкт-Петербург, 2018

*Настоящая программа не может быть использована другими подразделениями
университета и другими вузами без разрешения подразделения-разработчика программы.*

1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Алгоритмы и структуры данных», учебных ассистентов и студентов направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, обучающихся по бакалаврской программе «Прикладная математика и информатика» и изучающих дисциплину «Алгоритмы и структуры данных».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 01.03.02 «Прикладная математика и информатика» (уровень бакалавриата), утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 03.03.2017 №02.
- Основной профессиональной образовательной программой «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика»;
- Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2018 г.

2. Цели освоения дисциплины

Целью освоения дисциплины «Алгоритмы и структуры данных» является формирование у студентов теоретических знаний и практических навыков в области теории алгоритмов, современных структур данных и их реализации на языке программирования C++ для построения математических моделей дискретных структур и разработки программного обеспечения.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать методы оценки сложности алгоритмов в среднем и в худшем случаях, базовые и продвинутое абстрактные структуры данных, постановки основных задач, основные классы алгоритмов.
- Уметь оценивать сложность алгоритмов в среднем и в худшем случаях, выделять из практических задач их алгоритмическую составляющую, реализовывать изученные алгоритмы и структуры данных на процедурных языках программирования, выбирать оптимальные алгоритмы и структуры данных, в зависимости от конкретных ограничений на решение задачи, применять приближенные алгоритмы в тех случаях, когда эффективное точное решение невозможно.
- Иметь навыки (приобрести опыт) оценки сложности алгоритмов в среднем и в худшем случаях, реализации алгоритмов и структур данных на процедурных языках программирования.

В результате освоения дисциплины студент осваивает следующие компетенции:

| Компетенция | Код по ОС НИУ ВШЭ | Уровень формирования компетенции | Дескрипторы – основные признаки освоения (показатели достижения результата) | Формы и методы обучения, способствующие формированию и развитию компетенции | Форма контроля уровня сформированности компетенции |
|--|-------------------|----------------------------------|--|---|--|
| Способность учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной | УК-1 | РБ СД МЦ | Знает основные алгоритмы и структуры данных, применяемые для различных задач. Подбирает оптимальный алгоритм для конкретной практической задачи, анализирует его эффективность. Использует известные базовые алгоритмы и их модификации для решения разнообразных задач профессиональной деятельности. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, письменный экзамен |
| Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач | ОПК-2 | РБ СД МЦ | Знает основные способы модификации базовых алгоритмов, применяемые для различных задач. Подбирает необходимый и оптимальный алгоритм, способ его адаптации для конкретных практических задач. Использует базовые алгоритмы и подходы и модифицирует их, исходя из специфики решаемой задачи. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, письменный экзамен |
| Способен применять и модифицировать математические модели для решения задач в области | ОПК-3 | РБ СД | Формализует и описывает алгоритм решения поставленных практических задач. Математически корректно и | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная | Домашние задания, письменный экзамен |

| | | | | | |
|-------------------------------|--|----|--|--------|--|
| профессиональной деятельности | | МЦ | адекватно записывает алгоритмы, наиболее корректно описывающие дискретные объекты прикладной задачи. Реализует алгоритмы на языках программирования с использованием различных методологий программирования. | работа | |
|-------------------------------|--|----|--|--------|--|

4. Место дисциплины в структуре образовательной программы

Для образовательной программы «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика» настоящая дисциплина относится к дисциплинам базовой части блока дисциплин.

Основные положения данной дисциплины используются для освоения следующих дисциплин:

- Теория алгоритмов;
- Дополнительные главы алгоритмов и структур данных;
- Теория информации.

5. Тематический план учебной дисциплины

Курс рассчитан на 218 часов аудиторной нагрузки, из них 112 часов лекций и 106 часов практических занятий, общим объемом 11 зачетных единиц (418 часов).

| № | Название раздела | Всего часов | Аудиторные часы | | | Самостоятельная работа |
|---|---|-------------|-----------------|----------|----------------------|------------------------|
| | | | Лекции | Семинары | Практические занятия | |
| 1 | Основные понятия теории алгоритмов и структур данных | 74 | 18 | 0 | 18 | 38 |
| 2 | Динамическое программирование. Комбинаторные и графовые алгоритмы | 78 | 20 | 0 | 20 | 38 |
| 3 | Элементы теории сложности алгоритмов. Кратчайшие пути. Жадные алгоритмы | 78 | 20 | 0 | 20 | 38 |
| 4 | Деревья поиска, деревья отрезков и другие аналогичные структуры | 74 | 18 | 0 | 18 | 38 |
| 5 | Алгоритмы на графах. Потoki в орграфах. Строки | 58 | 18 | 0 | 16 | 24 |
| 6 | Игры на графах. Быстрое преобразование Фурье. Линейная алгебра | 56 | 18 | 0 | 14 | 24 |

| | | | | | |
|-------|-----|-----|---|-----|-----|
| ИТОГО | 418 | 112 | 0 | 106 | 200 |
|-------|-----|-----|---|-----|-----|

6. Содержание дисциплины

| | |
|---|--|
| <u>Раздел 1</u> Основные понятия теории алгоритмов и структур данных | |
| Тема 1 | <p>Простейшие алгоритмы и структуры данных. Асимптотика. Сумма гармонического ряда. Рекуррентные соотношения. Числа Фибоначчи. Сравнение асимптотик. Частичные суммы. Массив фиксированного размера. Список двусвязный, односвязный, на массиве. Вектор (расширяющийся массив). Стек, очередь, дек. Очередь и дек на циклическом массиве. Стек с минимумом. Очередь с минимумом через два стека. Разбор арифметического выражения со стеком за линейное время. Время работы: real, average, amortized. Схема с монетками. Примеры. Расширяющийся и сужающийся динамический массив. Доказательства времени работы. Вектор: избавление от амортизации. Очередь с минимумом: избавление от амортизации.</p> |
| Тема 2 | <p>Поиск и сортировки. Бинарный поиск. 3 версии. Средства языка C++. Вещественный бинарный поиск. Хранение множеств и мультимножеств в виде сортированных массивов. Два указателя. Средства языка C++. Алгоритм Мо. Два указателя на примере задачи «количество различных чисел на отрезке». Квадратичные сортировки: Insertion, Selection, Bubble. Сравнение. Стабильность. IntroSort. Оценка снизу на число сравнений. MergeSort. Нерекурсивная версия. QuickSort без доказательства. Две версии partition: на python, двумя указателями. Способы выбора элемента для partition. Сравнение сортировок за $O(n \log n)$. QuickSort. Доказательство по индукции с интегралами. QuickSort. Доказательство через дерево рекурсии и вероятность «сравнить два элемента». Порядковая статистика за линейное время, рандомизированный алгоритм. Средства C++. Порядковая статистика за линейное время, детерминированный алгоритм. Count Sort для чисел и для пар чисел, Radix Sort за $O(n \log n \cdot m)$. Bucket Sort. Две версии алгоритма. Три теоремы о времени работы. Kirkpatrick Sort за $O(n \log \log C)$</p> |
| Тема 3 | <p>Хэш-таблицы и кучи. Хэш-таблица на списках. Хэш-таблица с открытой адресацией. Rehash. Сравнение двух реализаций хэш-таблиц. Средства языка C++. Хэш-таблица: избавление от амортизации. Бинарная куча: хранение в массиве, Add, ExtractMin, DecreaseKey и обратные ссылки. Бинарная куча: построение за линию, heap sort. Средства языка C++. Аллокация памяти. Стек. Список. Кучи с хэш-таблицей. Оптимизация для C++. Преобразование операций. Ничего \rightarrow Del, Find \rightarrow Del, Add \rightarrow Merge. Пополняемые структуры данных: Build \rightarrow Add, Del. Пополняемый массив. Van Embde Boas trees. Добавление элемента и извлечение минимума. MinMax Heap. Подробная оценка времени работы. Leftist Heap, Skew Heap. d-куча. Куча, умеющая {Add, Min, Merge, DecreaseKey} за $O(1)$, ExtractMin за $O(n)$. Нижняя оценка на время построения бинарной кучи. Улучшение Add $O(\log n) \rightarrow O(1)$ и Merge $O(\log n) \rightarrow O(1)$ (bootstrapping). Binomial Heap. Fibonacci Heap: Add и Merge за $O(1)$, алгоритм для DecreaseKey. Fibonacci Heap: доказательство корректности, оценка времени работы.</p> |
| <u>Раздел 2</u> Динамическое программирование. Комбинаторные и графовые алгоритмы. | |
| Тема 1 | <p>Динамическое программирование. Вперёд, назад, ленивая. Восстановление ответа. Пример: «калькулятор». Ациклический граф состояний. Формулировка задачи в терминах графа. Формулировка терминов «вперёд», «назад», «ленивая» через граф состояний. Задача про путь на матрице с дырками. Min путь, max путь, кол-во путей. Рюкзак без стоимостей за $O(nW)$ времени,</p> |

| | |
|--|--|
| | <p>$O(W)$ памяти с восстановлением ответа. Применение bitset. Проблемы с восстановлением ответа в случае стоимостей. Задача НОП. Восстановление ответа без дополнительных ссылок назад. Оптимизация памяти, если нет восстановления ответа. Оптимизация памяти для восстановления ответа в НОП с помощью bitset. Алгоритм Хиршберга для НОП за $O(n^2)$ времени, $O(n)$ памяти с восстановлением ответа. Трёхсторонний путь по матрице мин веса (про министерство). Оптимизация памяти до $O(n \sqrt{2} m)$. Задача LCP. Задача «расстояние Левенштейна». Задача LIS (НВП). Решения за $O(n^2)$ и за $O(n \log n)$. Измельчение перехода. Задача про погрузку грузов на корабль, решения за $O(n^4)$, $O(n^3)$, $O(n^2)$. Возведение матрицы в степень: количество путей длины k, решение линейного рекуррентного соотношения. Два указателя на примере задачи про выбор k точек на прямой среди n данных. Решение задачи про выбор k точек на прямой среди n данных методом «разделяй и властвуй». Подотрезки. Расстановка скобок в выражении, умножение матриц. Восстановление ответа. Поддержка. Максимальное по весу паросочетание в дереве.</p> |
| Тема 2 | <p>Комбинаторика и подмножества. Комбинаторика. Следующий и предыдущий объект. Перестановки, скобочные последовательности. Объект по номеру, номер по объекту. Перестановки, скобочные последовательности. Подсчет числа разбиений на слагаемые: упорядоченные, неупорядоченные. Операции с множествами, как с n-битовыми числами. Количество бит в числе за $O(2^n)$. Сумма на подмножестве за $O(2^n)$. Гамильтонов путь и цикл за $O(2^{n \cdot n})$. Перебор подмножеств и надмножеств. Покраска вершин графа в минимальное число цветов за $O(3^n)$. Оптимизация покраски вершин графа в минимальное число цветов с $O(3^n)$ до $O(2.44^n)$. Set Cover («рюкзак на подмножествах»). Meet in the Middle. Подсчет числа клик за $O(2^{n/2})$. Meet in the Middle. Рюкзак с ценами за $O(2^{n/2} \cdot n)$. Динамика по скошенному профилю: количество замощений доминошками за $O(2^{h \cdot h \cdot w})$. Только рекурсивная реализация.</p> |
| Тема 3 | <p>Графы и dfs. База. Хранение графов. Матрица смежности, bitset[], vector[], set[], мультисписок. dfs. Def: путь, простой путь. Поиск простого пути между a и b dfs-ом за $O(E)$. Def: связность, сильная связность, слабая связность, компоненты связности. Поиск компонент связности за $O(V + E)$. Циклы. Def: цикл, контур. Поиск цикла в орграфе, в неорграфе за $O(V + E)$. Проверка на двудольность. Поиск topsort двумя способами. Связь с динамическим программированием. Выделение компонент сильной связности за $O(V + E)$. Конденсация графа. Euler. Критерий Эйлеровости. Поиск Эйлерова пути и цикла за $O(V + E)$. Задачи: дополнение графа до Эйлерова, разбиение рёбер графа на минимальное число путей. 2-connectivity. Поиск мостов. Выделение компонент реберной двусвязности. 2-connectivity. Поиск точек сочленения. Выделение компонент вершинной двусвязности. 2-SAT. Решение за $O(n + m)$. Задачи: 2-List-Coloring.</p> |
| <p><u>Раздел 3</u> Элементы теории сложности алгоритмов. Кратчайшие пути. Жадные алгоритмы</p> | |
| Тема 1 | <p>Сложность и рандом. Неразрешимость halting problem. Разрешимые: decision, search. Язык. Классы: DTime, P, EXP. Иерархия по времени. $P \neq EXP$. Классы NP, coNP, coNEXP. Примеры: k-CLIQUE, MAX-CLIQUE, HAM-PATH, PRIME, coPRIME, IS-SORTED. Классы NP-hard, NP-complete, полиномиальное сведение. $BH \in NP$-complete. Сведения $BH \rightarrow CIRCUIT-SAT \rightarrow SAT \rightarrow 3-SAT$. Сведения $3-SAT \rightarrow k-IND \rightarrow k-CLIQUE \rightarrow VERTEX-COVER$. Задачи максимизации; search \rightarrow decision; search 3-SAT \rightarrow search k-IND. Решение NP задач. Классы: RP, coRP, ZPP. Сравнение с NP. Понижение ошибки. Задачи: поиск квадратичного невычета, проверка на простоту (Ферма, Миллер-Рабин), поиск частого числа, 3-LIST-COLORING, matrix multiplication testing. $ZPP = RP \cap coRP$ и вложение классов. BPP и понижение ошибки. Применение случайных чисел: кодирование, подсчёт суммы чисел без разглашения. Парадокс дней рождений с доказательством. Приближённые алгоритмы для MAX-3-SAT. Факторизация чисел. p-эвристика</p> |

| | |
|--|--|
| | <p>Полларда, версия с $O(n^{1/4})$ арифметическими операциями. Решения 3-SAT: детерминированный алгоритм 1.73^n, рандомизированный за 1.5^n (с доказательством). Лемма Шварца-Зиппеля (формулировка). Матрица Татта и поиск совершенного паросочетания. Random shuffle массива. Игра на 0-1-дереве, min-max-игра на дереве. Квадратный корень по простому модулю за $O(\log p)$.</p> |
| Тема 2 | <p>Кратчайшие пути. bfs. Слоистая версия, версия с очередью. Кратчайший путь за $O(V + E)$. Сравнение с dfs. Матрица расстояний за $O(V E)$. Граф кратчайших путей. 0-1-bfs. Две версии целочисленного 1-k-bfs-a. Вещественный 1-k-bfs и A-B-bfs. Вещественный 0-1-bfs. Версия за $O(V^2 w)$. Версия для 1-k-графа за $O(E \log k)$, $O(E \log \log(V k))$. Дейкстра. SSSP. APSP. Кратчайший путь за $O(V^2)$, $O(E \log V)$, $O(E + V \log V)$, $O(E \log E / V)$. A*. Доказательство корректности для графов с неравенством треугольника. Сравнение скорости с Дейкстрой. Пример, на котором A* даёт преимущество. Флойд. Решение за $O(V^3)$. Восстановление пути. Транзитивное замыкание за $O(V^3 w)$. Форд-Беллман. Решение за $O(V E)$. Реализация на одномерном массиве. Восстановление пути. Оптимизации: break, queue, random_shuffle. Джонсон. Идея потенциалов. Матрица расстояний за $O(V E + V^2 \log V)$. Отрицательные циклы. С помощью Флойда. С помощью Форд-Беллмана. Восстановление ответа, доказательство корректности. Гольдберг. Основная идея. Лечение одной вершины за $O(V + E)$. Вывод асимптотики $E \sqrt{V \log N}$. Лечение пачки из хотя бы \sqrt{k} вершин за $O(V + E)$. Min mean cycle за $O(V E \log nC)$. Карп. Min mean cycle за $O(V E)$. Йен. Поиск k-го простого кратчайшего пути за $O(kV \cdot \text{Dijkstra})$.</p> |
| Тема 3 | <p>DSU, MST, Жадности, Приближённые алгоритмы, Центроид. DSU на списках. Задача: для каждой пары вершин найти путь с минимальным max ребром. DSU на деревьях: эвристика сжатия путей, две версии ранговой эвристики. Доказательство того, что любая одна из трёх эвристик даёт время работы $O(\log n)$. MST. Лемма о разрезе. Краскал. Прим. Сравнение скорости. Доказательства. MST. Борувка. Время работы $O(E \log(V^2/E))$, доказательство. Approx. Salesman: 2-OPT, 1.5-OPT, \nexists C-OPT алгоритма для графов без неравенства треугольника. Жадность. Гамильтонов путь. Правило Варнсдорфа. Переборное решение для произвольного графа. Approx. $\ln n$-OPT для SET-COVER. Реализация за линию. 2-OPT для VERTEX-COVER. Задача о надстроке. Жадность. Хаффман. Безпрефиксные коды. Поиск кодов, кодирование, декодирование. Поиск кодов по массиву частот за $O(\text{sort} + \Sigma)$. Способы хранения дерева кодов. Задачи на сортировку. Способы доказательства, вывод компаратора. Примеры: непрерывный рюкзак, файлы на ленте, задачи с дедлайнами (3 версии), ещё 1 любая задача. Задания с дедлайнами: выполнить все за $O(n \log n)$, выполнить максимум за $O(n^2)$. Задания с дедлайнами: выполнить максимум. Решение за $O(n^2)$ и $O(n \log n)$. Задача о двух станках. Рюкзаки. Partition. Решения: жадное, алгоритм Кармаркар-Карпа, PTAS. Knapsack. Неприближаемость. Решения: жадность, PTAS, FPTAS. Bin packing. Неприближаемость. Решения: FF (2-OPT), FFD (11/9-OPT), решение для случая «$a_i \geq \epsilon$, различных $\leq K$». Bin packing. PTAS: сведение к случаю «$a_i \geq \epsilon$, различных $\leq K$». Центроидная декомпозиция: построение, минимум на пути. Центроидная декомпозиция: покраска вершин на расстоянии $\leq d$.</p> |
| <p><u>Раздел 4</u> Деревья поиска, деревья отрезков и другие аналогичные структуры.</p> | |

| | |
|---|--|
| Тема 1 | <p>Деревья поиска. BST. Определение, add/del/prev/next/lower_bound, использование списка и хеш-таблицы. Обработка равных ключей. Персистентность. Персистентные операции “добавление” и “удаление”. AVL-дерево. Инвариант, single/double rotation, add/del. Число вращений при add/del. Split/Merge. Persistent AVL. Неявный ключ. B-дерево: find, add/del, split/merge. B* -дерево и B+-дерево. Неявный ключ. 2-3-дерево, 2-3-4-дерево, RB-дерево, AA-дерево. Неявный ключ. RBST (без y), Treap (с y). Split/merge. Два способа для add/del. Персистентная версия. Splay-дерево: add/del, split/merge, потенциал, формулировки теорем. Splay-дерево: доказательство общей теоремы о времени работы и теоремы о времени работы с учётом частот. Запрос на отрезке, модификация на отрезке, reverse на отрезке.</p> |
| Тема 2 | <p>Деревья отрезков и другие структуры Центроидная декомпозиция: построение, минимум на пути. Покраска вершин на расстоянии $\leq d$, число путей длины $\leq d$. Rope: интерфейс, возможные реализации. Skip-List: add/del, split/merge. Sqrt декомпозиция. По массиву (с примером), по запросам (с примером). Split/rebuild (с примером), split/merge (с примером), выбор k. Sqrt декомпозиция по запросам. Dynamic sorted array. Dynamic Connectivity Offline. Персистентность. Подход к любой структуре (примеры: СНМ, хеш-таблица); offline-решение. Персистентные стек и очередь с операциями за $O(1)$. Персистентная дек с операциями за $O(\log n)$. Персистентность через fat-nodes. Применение в вычислительной геометрии. ДО. С операциями снизу (все оценки, код). С операциями сверху (все оценки, сравнение с реализацией снизу). Динамическое; сжатие координат; пример использования обеих идей в scanline. Многомерные деревья: сортированный массив, treap, ДО, 2D ДО. Scanline: число точек в прямоугольнике, число прямоугольников, покрывающих точку. Персистентная версия scanline для обеих задач. Площадь объединения прямоугольников.</p> |
| Тема 3 | <p>LCA, RMQ. RMQ. Формулировка задачи. Lower bound. Решение через ДО, корневую, Sparse Table. Замыкание Sparse Table до $\langle n \log^* n, \log^* n \rangle$. Disjoint Sparse Table. LCA. Формулировка, решение двоичными подъёмами, функция на пути дерева. RMQ± 1 за $\langle O(n), O(1) \rangle$. LCA за $\langle O(n), O(1) \rangle$. Три типа эйлерового обхода, пример применения каждого. RMQ за $\langle O(n), O(1) \rangle$. Построение декартова дерева за $O(n)$. LCA в offline. Алгоритм Тарьяна. LA. Решение в offline. Алгоритм Вишкина. LA. Реализация Вишкина за $\langle O(n \log n), O(1) \rangle$. Решение за $\langle O(n), O(1) \rangle$. Euler Tour Tree. Функция от поддерева, модификация в поддереве. Heavy Light Decomposition. Функция на пути дерева, модификация на пути дерева. Link Cut Tree. Описание структуры, введение потенциала, оценка Expose. Оценка MakeRoot, Link, Cut. Оценка Expose и MakeRoot со Splay-деревом. MST за $O(n + m)$. Оценка $\min(m \log n, n^2)$ в худшем. RMQ в offline через алгоритм Тарьяна.</p> |
| <p><u>Раздел 5</u> Алгоритмы на графах. Поток в орграфах. Строки</p> | |

| | |
|---|---|
| Тема 1 | <p>Паросочетания, раскраски.</p> <p>Matching. Определения и сложность задач в двудольном и произвольном графе: паросочетание, вершинное покрытие, независимое множество, совершенное паросочетание. Матрица Татта (без док-ва). Лемма о дополняющем пути. Алгоритмы поиска паросочетания: обычный, Куна. Оптимизации алгоритма Куна: не чистить пометки, вообще не чистить пометки, жадная инициализация, быстрое обнуление. Теорема Кёнига, поиск вершинного покрытия за $O(V + E)$. Применение Куна для поиска паросочетания в произвольном графе. Классификация рёбер двудольного графа по принадлежности паросочетанию. Разбиение вершин орграфа на циклы. Разбиение вершин ациклического орграфа на минимальное число путей. Теорема Дилворта, поиск максимальной антицепи. Stable (marriage problem). Алгоритм за $O(E)$. Венгерский алгоритм. Задача о назначениях. Покраски. Рёберная. Теорема Визинга. Связь с паросочетанием. Покраска рёбер двудольного графа. Алгоритмы рёберной покраски регулярного двудольного графа за $O(\text{Matching} \cdot \log D)$. Алгоритмы рёберной покраски двудольного графа за $O(E^2)$. Вершинные. Брукс, практически ценный алгоритм.</p> |
| Тема 2 | <p>Потоки</p> <p>Потоки. Def: поток, разрез, величина потока, величина разреза, остаточная сеть, циркуляция, прямые и обратные рёбра, декомпозиция потока. 17. (a) Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона. Поиск min разреза по max потоку. Декомпозиция потока за $O(E^2)$, $O(V E)$. k рёберно и вершинно непересекающихся путей в орграфе и неорграфе. Поиск паросочетания, связь контролирующего множества с разрезом. Алгоритм Эдмондса-Карпа. Существование max потока и целочисленного max и потока. Алгоритм масштабирования потока. Алгоритм Диница и его спаривание с масштабированием. Алгоритм Хопкрофта-Карпа и две теоремы Карзанова. [L, R]-циркуляция, [L, R]-flow, [L, R]-max-flow. Глобальный разрез: алгоритм Штор-Вагнера без доказательства. Диниц + link-cut. Глобальный разрез: алгоритм Каргера-Штейна. Mincost k-flow, mincost max flow, mincost circulation, транспортная задача. Формулировки задач. Критерий оптимальности mincost k-flow. Алгоритмы построения mincost k-flow и mincost flow через дополняющие пути в графе без отрицательных циклов. Mincost k-flow. Дейкстра и потенциалы. Графы с отрицательными циклами. Алгоритм Клейна. Полиномиальная версия (без док-ва времени работы). Capacity scaling. Псевдополиномиальный алгоритм для mincost k-flow.</p> |
| Тема 3 | <p>Строки.</p> <p>Z-функция. Поиск подстроки в строке и периода строки. Префикс функция. Поиск подстроки в строке и периода строки. Алгоритмы Боэра-Мура поиска подстроки в строке. Хеши: полиномиальный хеш, хеш подстроки, Алгоритм Рабина-Карпа поиска подстроки в строке с $O(1)$ доппамяти. Хеши: вероятности. 3 леммы. Каким хешом пользоваться? Обоснование вероятности коллизии в худшем и в среднем. Хеши: антихеш тесты. Строка Гуэ-Морса. Хеши: поиск общей подстроки, поиск LCP, построение суффмассива за $O(n \log^2 n)$. Палиндромы. Подсчет числа за $O(n \log n)$ хешами и $O(n)$ алгоритмом Манакера. Самый длинный за $O(n)$. Суффмассив: построение за $O(n \log n)$. Поиск подстроки в тексте. 41. Алгоритм Касаи за $O(n)$. Построение за $O(n)$, Каркайнен-Сандерс. Поиск строки в тексте за $O(S + \log T)$. Бор, сжатый бор, хранение. Построение суффдерева за $O(n^2)$ двумя способами. Подстрока в тексте. Автомат Ахо-Корасик. Поиск словарных слов в тексте. Суффдерево: алгоритм Укконена построения за $O(n)$. Преобразование суффдерева \leftrightarrow (суффмассива + LCP) за $O(n)$. Решение задач суффмассивом и суффдеревом: поиск общей подстроки, число различных подстрок.</p> |
| <p><u>Раздел 6</u></p> <p>Игры на графах. Быстрое преобразование Фурье. Линейная алгебра.</p> | |

| | |
|--------|--|
| Тема 1 | Игры на графах и теория чисел. Игры на графах. Ациклический граф. Граф с циклами. Ретро-анализ. Длина игры. Функция Гранди. Прямая сумма игр. Эквивалентность игр. Подсчет через тех. Сумма игр через XOR. Задачи на Гранди: задача про скамейки, hackenbush. ТЧ. Решето Эратосфена. Версия за $O(n \log \log n)$, время работы, оптимизация в константу раз. Версия за $O(n)$. Подсчет мультипликативных функций на $[1, n]$ за $O(n)$: ϕ , число делителей, сумма делителей. Расширенный алгоритм Евклида, способы обращения числа. Время работы. Поиск обратных к $1..k$ за $O(k)$. RSA. Алгоритм кодирования, декодирования, сложность вычислений. Взлом в частных случаях. Дискретный логарифм за $O(\sqrt{p})$. Протокол Диффи—Хеллмана. Первообразный корень: проверка, поиск (без док-ва вероятности попадания). Применение: корень k -й степени по простому модулю – сведение к делению. Использование в длинной арифметике. |
| Тема 2 | Линейная алгебра и четыре русских. Гаусс для полей R, F_p за $O(n^3)$. Решение для треугольной матрицы за $O(n^2)$. Реализация Гаусса для невырожденной квадратной матрицы. Гаусс в F_2 за $O(n^3/w)$. Реализация Гаусса для произвольной матрицы $m \times n$. Свободные переменные. Базис решений. Разложение вектора в базисе. Гаусс и погрешность. Матрица Гильберта. Методы борьбы с погрешностью. Простой метод итераций. Вычисление определителя и обратной матрицы за $O(n^3)$. Вероятностные задачи на графах – вычисление вероятностей и матожиданий. Разложение вектора в базисе. Проверка принадлежности точки параллелепипеду. Расстояние от точки до подпространства. Четыре русских. Умножение матриц над F_2 за $O(n^3/(w \log n))$. НОП над константным алфавитом за $O(n^2/\log n)$. |
| Тема 3 | Быстрое преобразование Фурье. База: извлечение корня из комплексного числа, единственность интерполяционного многочлена. Схема умножения многочленов за $O(n \log n)$. Интерполяция за $O(n^2)$ (см. практику). Рекурсивное вычисление DFT за $O(n \log n)$ (над комплексными числами). Нерекурсивная оптимизированная реализация за $O(n \log n)$, обратная битовая запись за $O(n)$. Связь обратного DFT и прямого DFT. Вычисление обратного за $O(n \log n)$. Целочисленное DFT. Оптимизация: два вещественных DFT в одном комплексном. Ограничение на коэффициенты для комплексного случая. Умножение чисел за $O(n \log n)$, выбор системы счисления. Возведение в степень. Поиск с ошибками. Поиск с ошибками и шаблоном. (см. практику) Длинная арифметика: хранение, линейные операции, деление чисел за $O(n^2/w^2)$. Gcd за $O(n^2/w^2)$. Деление чисел и многочленов/чисел за $O(n \log^2 n)$ методом разделяй и властвуй. |

7. Оценочные средства

7.1. Формы контроля знаний студентов

| Тип контроля | Форма контроля | 1 год | | | | 2 год | | Параметры |
|--------------|---------------------|----------|----------|----------|----------|----------|----------|-----------------------------|
| | | 1 модуль | 2 модуль | 3 модуль | 4 модуль | 1 модуль | 2 модуль | |
| Текущий | Домашнее задание №1 | * | | | | | | Письменное домашнее задание |
| | Домашнее задание №2 | | * | | | | | Письменное домашнее задание |
| | Домашнее задание №3 | | | * | | | | Письменное домашнее задание |

| | | | | | | | | |
|---------------|-----------------------|---|---|---|---|---|---|-----------------------------|
| | Домашнее задание №4 | | | | * | | | Письменное домашнее задание |
| | Домашнее задание №5 | | | | | * | | Письменное домашнее задание |
| | Домашнее задание №6 | | | | | | * | Письменное домашнее задание |
| Промежуточный | Письменный экзамен №1 | * | | | | | | Экзамен в письменной форме |
| | Письменный экзамен №2 | | * | | | | | Экзамен в письменной форме |
| | Письменный экзамен №3 | | | * | | | | Экзамен в письменной форме |
| | Письменный экзамен №4 | | | | * | | | Экзамен в письменной форме |
| | Письменный экзамен №5 | | | | | * | | Экзамен в письменной форме |
| Итоговый | Письменный экзамен №6 | | | | | | * | Экзамен в письменной форме |

7.2. Критерии и шкалы оценки, примеры заданий

7.2.1. Текущий контроль

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

ДОМАШНЕЕ ЗАДАНИЕ №1

Домашнее задание №1 выдается студентам в одном варианте и состоит из 6 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №1:

1. **Параллельный минимум и максимум** [максимум 2 балла за правильно выполненную задачу]. Дан массив из $2n$ чисел. Найти минимальное и максимальное за $3n - 2$ сравнения.
2. **Поиск статистики.** [максимум 5 баллов за правильно выполненные задачи] Даны m сортированных массивов длины n . Нужно без дополнительного подсчета найти $-ю$ порядковую статистику. Числа от 0 до MAX.
 - 2.1. [2 балла] За $O(m + k \log m)$.
 - 2.2. [2 балла] За $O(m \log \text{MAX} \log n)$.

- 2.3. [1 балл]: вероятностное $\mathcal{O}(m \log^2 poly(n, m))$.
3. **Ближайший по значению** [максимум 1 балл за правильно выполненную задачу]. Даны отсортированные массивы a и b длины n . Для каждого элемента a найти ближайший по значению элемент b . $\mathcal{O}(n)$.
4. **В этом задании требуется прислать код на языке C/C++!** [максимум 2 балла за правильно выполненные задачи]. Множество и мультимножество можно хранить в виде отсортированного массива. Даны два множества A и B в отсортированном виде, за $\mathcal{O}(|A| + |B|)$ построить в таком же виде их
- 4.1. [1 балл] множество-разность. Пример: $\{1, 2, 3\} \setminus \{2, 4\} = \{1, 3\}$.
- 4.2. [1 балл] множество-объединение. Пример: $\{1, 2, 3\} \cup \{2, 4\} = \{1, 2, 3, 4\}$.
5. **Ближайший по координате** [максимум 2 балла за правильно выполненную задачу]. Даны $n \leq 10^6$ точек на плоскости, координаты целые до 10^9 по модулю. Приходят $q \leq 10^6$ запросов: дана точка p , найти для нее ближайшие по X слева и справа точки с таким же Y и ближайшие по Y точки с таким же X сверху и снизу. Запросы online, нужно на каждый запрос отвечать сразу, а не на все вместе в конце.
6. **Отложенные операции** [максимум 2 балла за правильно выполненную задачу]. Рассмотрим структуру данных, хранящую мультимножество целых чисел S и умеющую обрабатывать два запроса: \cdot $\text{count}(S \cap [L..R])$ (количество элементов от L до R) \cdot $\text{add}(x)$ Структура устроена так: храним отсортированный массив a и массив b в произвольном порядке, при этом поддерживаем $|b| \leq \sqrt{|a|}$. На запросы count отвечаем, делая бинпоиск в a и линейный проход в b . При добавлении элемента добавляем его в конец b , если после этого $|b| > \sqrt{|a|}$, то $\{a = \text{merge}(a, \text{sort}(b)), b = []\}$. Найти и доказать амортизированное время обработки запросов.

Критерии оценивания и шкала оценки домашнего задания №1

| Оценка | Критерии выставления оценки |
|--------------------------------|-------------------------------------|
| «Отлично» (8-10) | Решено задач на 12 или более баллов |
| «Хорошо» (6-7) | Решено задач на 9-11 баллов |
| «Удовлетворительно» (4-5) | Решено задач на 6-8 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 6 баллов |

ДОМАШНЕЕ ЗАДАНИЕ №2

Домашнее задание №2 выдается студентам в одном варианте и состоит из 6 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №2:

1. k -е разбиение в другом порядке [максимум 2 балла за правильно

- выполненную задачу]. Найдите -е лексикографически разбиение числа n на возрастающие слагаемые. $\mathcal{O}(n^2)$. Разбиение – вектор $a_1 < a_2 < \dots < a_k$.
- Странные числа [максимум 3 балла за правильно выполненную задачу]. Даны k, L, R, A, B . Найдите -е по величине число среди таких, что выполнены все три свойства: а) $A \leq x \leq B$ б) В записи числа используются только нечётные цифры в) Если прочитать число задом наперёд, то $L \leq \text{reverse}(x) \leq R$. Ограничения: $1 \leq k, A, B, L, R \leq 1018$.
 - Подмножества надмножеств подмножеств [максимум 1 балл за правильно выполненную задачу]. Оцените сложность кода: `for (a = 0; a < (1 << n); ++a) for (b = a; b < (1 << n); b = (b + 1) | a) for (c = b; c > 0; c = (c - 1) & b);`
 - Циклы в неориентированном графе [максимум 3 балла за правильно выполненную задачу]. Дан граф. Для каждого подмножества вершин A проверить, есть ли простой цикл, проходящий по всем вершинам A ровно один раз (и только по вершинам из A). а) (2) $\mathcal{O}(2^n n)$. б) (3) $\mathcal{O}(2^n n)$.
 - Рюкзак в несколько заходов [максимум 2 балла за правильно выполненную задачу]. Есть n вещей, у каждой есть стоимость v_i и вес w_i . Есть рюкзак, в котором можно унести набор вещей суммарного веса не более W за один подход. За $m = 2k$ подходов унести вещи максимальной суммарной стоимости. Время $\mathcal{O}(3nk)$. (+1) балл, если работает не только для m – степеней двойки.
 - Set Cover [максимум 3 балла за правильно выполненную задачу]. Даны $A, B_1, \dots, B_m \subseteq \{0, \dots, n-1\}$. Выбрать минимальный набор $\{B_{ij}\}: \cup B_{ij} = A$. а) (1) $\mathcal{O}(2^m)$. б) (2) $\mathcal{O}(2^{nm})$.

Критерии оценивания и шкала оценки домашнего задания №2

| Оценка | Критерии выставления оценки |
|--------------------------------|-------------------------------------|
| «Отлично» (8-10) | Решено задач на 12 или более баллов |
| «Хорошо» (6-7) | Решено задач на 9-11 баллов |
| «Удовлетворительно» (4-5) | Решено задач на 6-8 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 6 баллов |

ДОМАШНЕЕ ЗАДАНИЕ №3

Домашнее задание №3 выдается студентам в одном варианте и состоит из 6 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №3:

- Задания и дедлайны [максимум 2 балла за правильно выполненную задачу]. Вы – студент, у вас есть куча домашних заданий. У каждого задания есть t_i – сколько часов его нужно делать. Чтобы -е задание зачли, его нужно начать делать не позже дедлайна d_i . После того, как вы начали делать задание, его обязательно довести до конца.

- 1.1 Можно ли выполнить все задания в срок?
- 1.2 Сколько максимум заданий можно выполнить в срок?
2. Хаффман: у частых символов короткие коды [максимум 2 балла за правильно выполненную задачу]. Доказать, что если есть символ частоты > 2.5 , то есть символ с длиной кода один.
3. Хаффман и длины кодов [максимум 2 балла за правильно выполненную задачу]. Если длина текста n , какова максимальная длина кода одного символа?
4. Сбалансированное разделение [максимум 2 балла за правильно выполненную задачу]. Под «поддеревом» в данной задаче мы понимаем связный подграф дерева. Дано дерево, разбить все его рёбра на два поддерева, пересекающиеся по одной вершине. Размеры поддеревьев должны отличаться не более чем в 2 раза.
5. Авторитеты [максимум 3 балла за правильно выполненную задачу]. Есть n человек. Человек i готов примкнуть к нашей коалиции, если наш авторитет хотя бы a_i , при этом он изменит наш авторитет на b_i . Наш изначальный авторитет равен A . а) (2) Можем ли завербовать всех людей? б) (1) Какое максимальное число людей мы можем завербовать? $O(n^2)$
6. Составь свою ППСП [максимум 2 балла за правильно выполненную задачу]. Даны n строк, состоящих из круглых скобок. Строка s называется ППСП (почти правильная скобочная последовательность), если в конец s можно дописать несколько закрывающих скобок, чтобы s стала правильной. Данные строки можно склеивать в произвольном порядке. Суммарную длину строк обозначим L . а) (1) Можно ли построить ППСП, используя все строки? $O(L + n \log n)$ б) (1) Выберите строки тах суммарной длины, чтобы получилась ППСП. $O(L+n^2)$. в) (+1) Выберите максимальное число строк, чтобы получилась ППСП. $O(L + n \log n)$. г) (+2) Выберите строки максимальной суммарной длины. $O(L \log n)$.

Критерии оценивания и шкала оценки домашнего задания №3

| Оценка | Критерии выставления оценки |
|--------------------------------|-------------------------------------|
| «Отлично» (8-10) | Решено задач на 10 или более баллов |
| «Хорошо» (6-7) | Решено задач на 8-9 баллов |
| «Удовлетворительно» (4-5) | Решено задач на 6-7 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 6 баллов |

ДОМАШНЕЕ ЗАДАНИЕ №4

Домашнее задание №4 выдается студентам в одном варианте и состоит из 5 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №4:

1. STL база [максимум 2 балла за правильно выполненную задачу]. Каждый работник характеризуется своим уникальным `id` (`autoincrement`), именем, фамилией и зарплатой. Придумайте структуру данных для обработки следующих запросов. Разрешается использовать только стандартные STL-контейнеры.
 - 1.1. Добавить работника с данными именем, фамилией, зарплатой.
 - 1.2. Удалить работника с заданным `id`.
 - 1.3. По заданным имени/фамилии выведите сумму зарплат всех работников с таким именем/фамилией.
 - 1.4. Выведите `id`/фамилии/зарплаты всех работников с заданным именем.
 - 1.5. Выведите `id`/имена/зарплаты всех работников с заданной фамилией.
 - 1.6. Выведите `id`/имена/фамилии всех работников с заданной зарплатой.
 - 1.7. Выведите фамилии всех работников, зарплата которых находится в заданном диапазоне [`salary_min`, `salary_max`]. Границы – параметры запроса. Задание в том, чтобы выписать структуры, которыми вы хотите воспользоваться, а после объяснить, как ими пользоваться для ответов на запросы.
2. Персистентный СНМ [максимум 3 балла за правильно выполненную задачу]. а) (1.5) Online. Не забудьте оценить память и время работы. б) (1.5) Offline. Не забудьте оценить память и время работы.
3. Корень `RV` дерева [максимум 2 балла за правильно выполненную задачу]. В красно-черном дереве хранятся числа $\{1, 2, \dots, n\}$. Какие из этих чисел могут находиться в корне дерева?
4. В-дерево и работа руками [максимум 2 балла за правильно выполненную задачу]. Данные с диска читаются блоками размера ровно 4096 байт. Размер ключа равен 8 байт. Предложите конкретное k для B^* -дерева в предположении, что в дереве всегда будет $< 2^{32}$ вершин. Оцените глубину полученного дерева.
5. Маленькие числа на пути [максимум 2 балла за правильно выполненную задачу]. Дано дерево. У каждого ребра есть вес w_e и ценность $cost_e$. Запросы (v, u, W) : среди всех рёбер на пути v и u веса $\leq W$ выбрать ребро \max стоимости.

Критерии оценивания и шкала оценки домашнего задания №4

| Оценка | Критерии выставления оценки |
|--------------------------------|------------------------------------|
| «Отлично» (8-10) | Решено задач на 9 или более баллов |
| «Хорошо» (6-7) | Решено задач на 7-8 баллов |
| «Удовлетворительно» (4-5) | Решено задач на 4-6 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 4 баллов |

ДОМАШНЕЕ ЗАДАНИЕ №5

Домашнее задание №5 выдается студентам в одном варианте и состоит из 6 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №5:

1. Самолёты [максимум 2 балла за правильно выполненную задачу]. Есть самолёты, у каждого самолёта есть время вылета t_i и вместимость $1 \leq k_i \leq 10$. Есть пассажиры, у каждого есть допустимый отрезок времён вылета $[l_i, r_i]$. Все самолёта и пассажиры летят/хотят из одного и того же пункта A в один и тот же пункт B . Нужно отправить максимальное число пассажиров.
2. Сведение [максимум 2,5 балла за правильно выполненную задачу]. Дан двудольный граф и чёрный ящик $\text{maxMatching}(\text{minVertexCover})$. Пусть чёрный ящик корректен и работает за $T(V, E)$. Найдите совершенное паросочетание за $O(T(V, E) + V + E)$. По сути мы доказываем, что задача «найти совершенное паросочетание» не сложнее, чем задача «по данному минимальному контролируемому множеству найти максимальное паросочетание».
3. Венгерский алгоритм [максимум 1,5 балла за правильно выполненную задачу]. Предъявите матрицу размера $n \times n$, $n = 3$, на которой $k = 2$ итераций внешнего цикла венгерского алгоритма найдут паросочетание размера k , не являющееся минимальным среди всех паросочетаний размера k . Ответ поясните. Венгерский алгоритм ищет дополняющие пути от вершин в порядке $0, 1, \dots, n - 1$.
4. Сделаем венгерский алгоритм полезнее [максимум 3 балла за правильно выполненную задачу]. а) Найдите максимальное по весу паросочетание произвольного размера. б) Найдите максимальное по весу паросочетание размера ровно k .
5. Подпоследовательности и gcd [максимум 3 балла за правильно выполненную задачу]. Разбить массив на подпоследовательности так, что сумма $\text{gcd}(a, b)$ по всем парам $\langle a, b \rangle$ соседних элементов всех последовательностей максимальна. $O(n^3)$. Каждый элемент массива должен попасть ровно в одну подпоследовательность.
6. ATSP [максимум 2 балла за правильно выполненную задачу]. Дан полный граф. Граф задан несимметричной матрицей неотрицательных весов. Задача maxATSP – найти в нём максимальный по весу гамильтонов цикл. Веса неотрицательны и удовлетворяют неравенству треугольника. Найдите 2-ОПТ приближение для maxATSP за полиномиальное время.

Критерии оценивания и шкала оценки домашнего задания №5

| Оценка | Критерии выставления оценки |
|---------------------|-------------------------------------|
| «Отлично» (8-10) | Решено задач на 11 или более баллов |
| «Хорошо» (6-7) | Решено задач на 8-10 баллов |

| | |
|--------------------------------|------------------------------------|
| «Удовлетворительно» (4-5) | Решено задач на 6-8 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 6 баллов |

ДОМАШНЕЕ ЗАДАНИЕ №6

Домашнее задание №6 выдается студентам в одном варианте и состоит из 5 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Пример домашнего задания №6:

1. Динамика по дереву [максимум 4 балла за правильно выполненную задачу]. Сколько способов вырезать из полного бинарного дерева глубины k поддерево размера s , содержащее корень исходного? Где здесь FFT?
2. Обратное по модулю [максимум 3 балла за правильно выполненную задачу]. Найти к числу x ($0 < x < m$) обратное по модулю m за $O(\log^2 m)$. Длинное m .
3. Циклические сдвиги [максимум 2 балла за правильно выполненную задачу]. Даны $A, B, |A| = |B| = n$. Найти D – такой циклический сдвиг B , что скалярное произведение A и D максимально.
4. Поиск подкартинки [максимум 3 балла за правильно выполненную задачу]. Даны две картины, заданные 256 оттенками серого. То есть даны матрицы целых чисел A и B . A по обоим размерам больше B . Найти такое наложение матрицы b на a , что суммарное квадратичное отклонение цветов минимально. То есть, найти такие i, j :

$$(a[x,y] - b[x+i,y+j])^2 \rightarrow \min_{x,y}$$
5. AVL деревья [максимум 2 балла за правильно выполненную задачу]. Дано: $n < 2^{16} = N = 2H$. И число $16 = H$. Найти количество AVL деревьев глубины h из n вершин по модулю $3 \cdot 2^{18} + 1$.
 - а) Используя вещественное FFT, за $O(NH^2)$.
 - б) Используя FFT по простому модулю, за $O(NH^2)$.

Критерии оценивания и шкала оценки домашнего задания №6

| Оценка | Критерии выставления оценки |
|--------------------------------|-------------------------------------|
| «Отлично» (8-10) | Решено задач на 11 или более баллов |
| «Хорошо» (6-7) | Решено задач на 8-10 баллов |
| «Удовлетворительно» (4-5) | Решено задач на 6-8 баллов |
| «Неудовлетворительно» (0-3) | Решено задач на менее чем 6 баллов |

7.2.2. Промежуточный контроль

Проверка качества освоения дисциплины производится в форме письменного экзамена.

ПИСЬМЕННЫЙ ЭКЗАМЕН №1

Письменный экзамен №1 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. Асимптотика. Обозначения o , O , Θ , Ω , ω . Основные свойства. Асимптотика многочлена.
2. Сумма гармонического ряда. Список делителей для всех чисел от 1 до n .
3. Рекуррентные соотношения: алгоритм Карацубы.
4. Рекуррентные соотношения: мастер-теорема (о простом рекуррентном соотношении).
5. Рекуррентные соотношения: теорема об экспоненциальном рекуррентном соотношении.
6. Числа Фибоначчи. Доказательства рекуррентностей по индукции.
7. Сравнение асимптотик: 5 терминов, теорема об их сравнении.
8. Шесть способов увеличить константу времени работы. Примеры. Кеш. Быстрые операции.
9. Частичный суммы. Массив фикс. размера. Список двусвязный, односвязный, на массиве.
10. Вектор (расширяющийся массив). Стек, очередь, дек. Средства языка C++ и их сравнение. Сравнение дека на списке и на массиве.
11. Очередь и дек на циклическом массиве. Стек с минимумом. Очередь с \min через два стека.
12. Очередь с минимумом, второй способ (дек минимумов).
13. Разбор арифметического выражения со стеком за линейное время.
14. Время работы: *real*, *average*, *amortized*. Схема с монетками. Примеры.
15. Расширяющийся и сужающийся динамический массив. Доказательства времени работы.
16. Вектор: избавление от амортизации. 2 способа.
17. Очередь с минимумом: избавление от амортизации.
18. Бинарный поиск. 3 версии. Средства языка C++.
19. Вещественный бинпоиск. Корни многочлена \forall степени. Определение границ бинпоиска.
20. Хранение множеств и мультимножеств в виде сортированных массивов. Два указателя. Средства языка C++.
21. Алгоритм Мо. Два указателя на примере задачи «количество различных чисел на отрезке».
22. Квадратичные: *Insertion*, *Selection*, *Bubble*. Сравнение. Стабильность. *IntroSort*.
23. Оценка снизу на число сравнений. Обобщения из практики.
24. *MergeSort*. Нерекурсивная версия. Число инверсий.
25. *QuickSort* без доказательства. Две версии *partition*: на *python*, двумя указателями. Способы выбора элемента для *partition*.
26. Сравнение сортировок за $O(n \log n)$.
27. *QuickSort*. Доказательство по индукции с интегралами.
28. *QuickSort*. Доказательство через дерево рекурсии и вероятность «сравнить два элемента».
29. Порядковая статистика за линейное время, рандомизированный алгоритм. Средства C++.

30. Порядковая статистика за линейное время, детерминированный алгоритм.
31. Count Sort для чисел и для пар чисел, Radix Sort за $O(n \log n m)$.
32. Bucket Sort. Две версии алгоритма. Три теоремы о времени работы.
33. Sort. Kirkpatrick Sort за $O(n \log \log C)$
34. Разделяй и властвуй: поиск двух ближайших точек.
35. Inplace алгоритмы. Unique. Rotate. Merge за $O(n \log n)$.
36. (a) DS. Хеш-таблица на списках.
36. Хеш-таблица на списках.
37. Хеш-таблица с открытой адресацией.
38. Rehash. Сравнение двух реализаций хеш-таблиц. Средства языка C++.
39. Хеш-таблица: избавление от амортизации.
40. Бинарная куча: хранение в массиве, Add, ExtractMin, DecreaseKey и обратные ссылки.
41. (b) DS. Бинарная куча: построение за линию, heap sort (inplace!). Средства языка C++.
42. Аллокация памяти. Стек. Список. Кучи с хеш-таблицей. Оптимизация для C++.
43. Преобразование операций. Ничего \rightarrow Del, Find \rightarrow Del, Add \rightarrow Merge.
44. Пополняемые структуры данных: Build \rightarrow Add, Del. Пополняемый массив.
45. Van Embde Boas trees. Добавление элемента и извлечение минимума.
46. MinMax Heap. Подробная оценка времени работы.
47. Leftist Heap, Skew Heap.
48. d-куча. Куча, умеющая {Add, Min, Merge, DecreaseKey} за $O(1)$, ExtractMin за $O(n)$.
49. (c) Heap. Нижняя оценка на время построения бинарной кучи.
50. Улучшение Add $O(\log n) \rightarrow O(1)$ и Merge $O(\log n) \rightarrow O(1)$ (bootstrapping).
51. Binomial Heap.
52. Fibonacci Heap: Add и Merge за $O(1)$, алгоритм для DecreaseKey.
53. Fibonacci Heap: доказательство корректности, оценка времени работы.

Критерии оценивания и шкала оценки письменного экзамена №1

| Оценка | Критерии выставления оценки |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

ПИСЬМЕННЫЙ ЭКЗАМЕН №2

Письменный экзамен №2 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. Динамическое программирование. Вперёд, назад, ленивая. Восстановление

ответа. Пример: «калькулятор». Ациклический граф состояний. Формулировка задачи в терминах графа. Формулировка терминов «вперёд», «назад», «ленивая» через граф состояний.

2. Задача про путь на матрице с дырками. Min путь, max путь, кол-во путей.
3. Рюкзак без стоимостей за $O(nW)$ времени, $O(W)$ памяти с восстановлением ответа. Применение bitset. Проблемы с восстановлением ответа в случае стоимостей. Решение рюкзака со стоимостями.
4. Задача НОП. Восстановление ответа без дополнительных ссылок назад. Оптимизация памяти, если нет восстановления ответа.
5. Оптимизация памяти для восстановления ответа в НОП с помощью bitset.
6. Алгоритм Хиршберга для НОП за $O(n^2)$ времени, $O(n)$ памяти с восстановлением ответа. Хиршберг для рюкзака.
7. Трёхсторонний путь по матрице мин веса (про министерство). Оптимизация памяти до $O(n^2 m)$.
8. Задача LCP. Задача «расстояние Левенштейна».
9. Задача LIS (НВП). Решения за $O(n^2)$ и за $O(n \log n)$.
10. Измельчение перехода. Задача про погрузку грузов на корабль, решения за $O(n^4), O(n^3), O(n^2)$.
11. Возведение матрицы в степень: количество путей длины k , решение линейного рекуррентного соотношения.
12. Два указателя на примере задачи про выбор k точек на прямой среди n данных. Без док-ва.
13. Решение задачи про выбор k точек на прямой среди n данных методом «разделяй и властвуй».
14. Подотрезки. Расстановка скобок в выражении, умножение матриц. Восстановление ответа.
15. Поддеревья. Максимальное по весу паросочетание в дереве.
16. Комбинаторика. Следующий и предыдущий объект. Перестановки, скобочные последовательности.
17. Комбинаторика. Объект по номеру, номер по объекту. Перестановки, скобочные последовательности.
18. Комбинаторика. Подсчет числа разбиений на слагаемые: упорядоченные, неупорядоченные.
19. Операции с множествами, как с n -битовыми числами. Количество бит в числе за $O(2n)$. Сумма на подмножестве за $O(2n)$.
20. Гамильтонов путь и цикл за $O(2^n n)$.
21. Проверка независимости всех подмножеств вершин за $O(2^n)$.
22. Перебор подмножеств и надмножеств. Покраска вершин графа в минимальное число цветов за $O(3^n)$.
23. Оптимизация покраски вершин графа в минимальное число цветов с $O(3^n)$ до $O(2.44^n)$.
24. Set Cover («рюкзак на подмножествах»).
25. Meet in the Middle. Подсчет числа клик за $O(2^{n/2})$.
26. Meet in the Middle. Рюкзак с ценами за $O(2^{n/2} n)$.
27. Динамика по скошенному профилю: количество замощений доминошками

- за $O(2hwh)$. Только рекурсивная реализация.
28. База. Хранение графов. Матрица смежности, $\text{bitset}\langle N \rangle$ [], $\text{vector}\langle \text{int} \rangle$ [], $\text{set}\langle \text{int} \rangle$ [], мультиспи-сок.
 29. Def: путь, простой путь. Поиск простого пути между a и b dfs-ом за $O(E)$.
 30. Def: связность, сильная связность, слабая связность, компоненты связности. Поиск компонент связности за $O(V + E)$.
 31. Циклы. Def: цикл, контур. Поиск цикла в орграфе, в неорграфе за $O(V + E)$. Проверка на двудольность.
 32. Поиск topsort двумя способами. Связь с динамическим программированием.
 33. Выделение компонент сильной связности за $O(V + E)$. Конденсация графа.
 34. Euler. Критерий Эйлеровости. Поиск Эйлера пути и цикла за $O(V + E)$. Задачи: дополнение графа до Эйлера, разбиение рёбер графа на минимальное число путей.
 35. 2-connectivity. Поиск мостов. Выделение компонент реберной двусвязности.
 36. 2-connectivity. Поиск точек сочленения. Выделение компонент вершинной двусвязности.
 37. 2-SAT. Решение за $O(n + m)$. Задачи: 2-List-Coloring.
 38. Матрица достижимости графа за $O(V E)$, за $O(V E)$.

Критерии оценивания и шкала оценки письменного экзамена №2

| Оценка | Критерии выставления оценки |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

ПИСЬМЕННЫЙ ЭКЗАМЕН №3

Письменный экзамен №3 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. Неразрешимость halting problem. Разрешимые: decision, search. Язык. Классы: DTime, P, EXP. Иерархия по времени. $P \neq EXP$.
2. Классы NP, coNP, coNEXP. Примеры: k-CLIQUE, MAX-CLIQUE, HAM-PATH, PRIME, coPRIME, IS-SORTED.
3. Классы NP-hard, NP-complete, полиномиальное сведение. $BH \in NP-complete$.
4. Сведения $BH \rightarrow \text{CIRCUIT-SAT} \rightarrow \text{SAT} \rightarrow 3\text{-SAT}$.
5. Сведения $3\text{-SAT} \rightarrow k\text{-IND} \rightarrow k\text{-CLIQUE} \rightarrow \text{VERTEX-COVER}$.
6. Задачи максимизации; search \rightarrow decision; search $3\text{-SAT} \rightarrow$ search $k\text{-IND}$.

Решение NP задач.

7. Классы: RP, coRP, ZPP. Сравнение с NP. Понижение ошибки.
8. BPP и понижение ошибки.
9. Применение случайных чисел: кодирование, подсчёт суммы чисел без разглашения.
10. Парадокс дней рождений с доказательством.
11. Приближённые алгоритмы для MAX-3-SAT.
12. Факторизация чисел. ρ -эвристика Полларда, версия с $O(n^{1/4})$ арифметическими операциями.
13. Решения 3-SAT: детерминированный алгоритм $1.73n$, рандомизированный за $1.5n$ (с доказательством).
14. Лемма Шварца-Зиппеля (формулировка). Матрица Татта и поиск совершенного паросочетания.
15. Random shuffle массива. Игра на 0-1-дереве, min-max-игра на дереве.
16. Квадратный корень по простому модулю за $O(\log p)$.
17. bfs. Слоистая версия, версия с очередью. Кратчайший путь за $O(V + E)$. Сравнение с dfs. Матрица расстояний за $O(V E)$. Граф кратчайших путей.
18. bfs. 0-1-bfs. Две версии целочисленного 1-k-bfs-a. Вещественный 1-k-bfs и A-B-bfs. Вещественный 0-1-bfs. bfs. Версия за $O(V^2)$. Версия для 1-k-графа за $O(E \log k)$, $O(E \log \log(V k))$.
19. Дейкстра. SSSP. APSP. Кратчайший путь за $O(V^2)$, $O(E \log V)$, $O(E + V \log V)$, $O(E \log E/V)$.
20. A*. Доказательство корректности для графов с неравенством треугольника.
21. Сравнение скорости с Дейкстрой. Пример, на котором A* даёт преимущество.
22. Флойд. Решение за $O(V^3)$. Восстановление пути. Транзитивное замыкание за $O(V^3)$. w
23. Форд-Беллман. Решение за $O(V E)$. Реализация на одномерном массиве. Восстановление пути.
24. Форд-Беллман. Оптимизации: break, queue, random_shuffle.
25. Джонсон. Идея потенциалов. Матрица расстояний за $O(V E + V^2 \log V)$.
26. Отрицательные циклы. С помощью Флойда. С помощью Форд-Беллмана. Восстановление ответа, доказательство корректности.
27. Гольдберг. Основная идея. Лечение одной вершины за $O(V + E)$. Вывод асимптотики
28. Гольдберг. Лечение пачки из хотя бы \sqrt{k} вершин за $O(V + E)$.
29. Min mean cycle за $O(V E \log nC)$.
30. Карп. Min mean cycle за $O(V E)$.
31. Йен. Поиск k-го простого кратчайшего пути за $O(kV \cdot \text{Dijkstra})$.
32. DSU на списках. Задача: для каждой пары вершин найти путь с минимальным max ребром.
33. DSU на деревьях: эвристика сжатия путей, две версии ранговой эвристики. Доказательство того, что любая одна из трёх эвристик даёт время работы $O(\log n)$.

34. DSU на деревьях: доказательство $O(m \log^* n)$
35. DSU на деревьях: доказательства $O(m + n \log^* n)$ (используя предыдущий билет).
36. DSU на деревьях: $O(m \log^{**} n)$, обратная функция Аккермана, доказательство $O((n + m)\alpha(n))$.
37. MST. Лемма о разрезе. Краскал. Прим. Сравнение скорости. Доказательства.
38. MST. Борувка. Время работы $O(E \log(V^2/E))$, доказательство.
39. Approx. Salesman: 2-OPT, 1.5-OPT, \exists C-OPT алгоритма для графов без неравенства треугольника.
40. Жадность. Гамильтонов путь. Правило Варнсдорфа. Переборное решение для произвольного графа.
41. Approx. In n-OPT для SET-COVER. Реализация за линию. 2-OPT для VERTEX-COVER. Задача о надстроке. Жадность. Хаффман. Безпрефиксные коды. Поиск кодов, кодирование, декодирование.
42. Поиск кодов по массиву частот за $O(\text{sort} + |\Sigma|)$. Способы хранения дерева кодов.
43. Жадность. Задачи на сортировку. Способы доказательства, вывод компаратора.
44. Примеры: непрерывный рюкзак, файлы на ленте, задачи с дедлайнами (3 версии), ещё 1 любая задача.
45. Жадность. Задания с дедлайнами: выполнить все за $O(n \log n)$, выполнить максимум за $O(n^2)$.
46. Жадность. Задания с дедлайнами: выполнить максимум. Решение за $O(n^2)$ и $O(n \log n)$.
47. Жадность. Задача о двух станках.
48. Рюкзаки. Partition. Решения: жадное, алгоритм Кармаркар-Карпа, PTAS.
49. Рюкзаки. Knapsack. Неприближаемость. Решения: жадность, PTAS, FPTAS.
50. Рюкзаки. Bin packing. Неприближаемость. Решения: FF (2-OPT), FFD (11/9-OPT), решение для случая « $a_i \geq \epsilon$, различных $\leq K$ ».
51. Рюкзаки. Bin packing. PTAS: сведение к случаю « $a_i \geq \epsilon$, различных $\leq K$ ».
52. Центроидная декомпозиция: построение, минимум на пути.
53. Центроидная декомпозиция: покраска вершин на расстоянии $\leq d$.

Критерии оценивания и шкала оценки письменного экзамена №3

| Оценка | Критерии выставления оценки |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

ПИСЬМЕННЫЙ ЭКЗАМЕН №4

Письменный экзамен №4 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. BST. Определение, add/del/prev/next/lower_bound, использование списка и хеш-таблицы. Обработка равных ключей.
2. BST. Персистентность. Персистентные операции “добавление” и “удаление”.
3. BST. AVL-дерево. Инвариант, single/double rotation, add/del. Число вращений при add/del.
4. BST. AVL-дерево. Split/Merge. Persistent AVL. Неявный ключ.
5. BST. B-дерево: find, add/del, split/merge. B* -дерево и B+ -дерево. Неявный ключ.
6. BST. 2-3-дерево, 2-3-4-дерево, RB-дерево, AA-дерево. Неявный ключ.
7. BST. RBST (без y), Treap (с y). Split/merge. Два способа для add/del. Персистентная версия.
8. BST. Splay-дерево: add/del, split/merge, потенциал, формулировки теорем.
9. BST. Splay-дерево: доказательство общей теоремы о времени работы и теоремы о времени работы с учётом частот.
10. BST. Запрос на отрезке, модификация на отрезке, reverse на отрезке.
11. Центроидная декомпозиция: построение, минимум на пути.
12. Центроидная декомпозиция: покраска вершин на расстоянии $\leq d$, число путей длины $\leq d$.
13. Rope: интерфейс, возможные реализации. Skip-List: add/del, split/merge.
14. Sqrt декомпозиция. По массиву (с примером), по запросам (с примером).
15. Sqrt декомпозиция. Split/rebuild (с примером), split/merge (с примером), выбор k.
16. Sqrt декомпозиция по запросам. Dynamic sorted array. Dynamic Connectivity Offline.
17. Персистентность. Подход к любой структуре (примеры: СНМ, хеш-таблица); offline-решение. Персистентные стек и очередь с операциями за $O(1)$.
18. Персистентная дек с операциями за $O(\log n)$.
19. Персистентность через fat-nodes. Применение в вычислительной геометрии.
20. ДО. С операциями снизу (все оценки, код).
21. ДО. С операциями сверху (все оценки, сравнение с реализацией снизу).
22. ДО. Динамическое; сжатие координат; пример использования обеих идей в scanline.
23. ДО. Многомерные деревья: сортированный массив, treap, ДО, 2D ДО.
24. ДО. Scanline: число точек в прямоугольнике, число прямоугольников, покрывающих точку. Персистентная версия scanline для обеих задач.
25. ДО. Площадь объединения прямоугольников.
26. ДО. k-я статистика на отрезке за $O(\log^3 n)$, $O(\log^2 n)$, $O(\log n)$.

27. Fractional-cascading, КД-дерево.
28. RMQ. Формулировка задачи. Lower bound. Решение через ДО, корневую, Sparse Table.
29. RMQ. Замыкание Sparse Table до $\langle n \log^* n, \log^* n \rangle$. Disjoint Sparse Table.
30. LCA. Формулировка, решение двоичными подъемами, функция на пути дерева. (
31. RMQ ± 1 за $\langle O(n), O(1) \rangle$.
32. LCA за $\langle O(n), O(1) \rangle$. Три типа эйлерового обхода, пример применения каждого.
33. RMQ за $\langle O(n), O(1) \rangle$. Построение декартова дерева за $O(n)$. (a) 35. LCA в offline. Алгоритм Тарьяна.
34. LA. Решение в offline. Алгоритм Вишкина.
35. LA. Реализация Вишкина за $\langle O(n \log n), O(1) \rangle$. Решение за $\langle O(n), O(1) \rangle$.
36. ETT. Euler Tour Tree. Функция от поддерева, модификация в поддереве.
37. HLD. Heavy Light Decomposition. Функция на пути дерева, модификация на пути дерева.
38. LCT. Link Cut Tree. Описание структуры, введение потенциала, оценка Expose.
39. LCT. Link Cut Tree. Оценка MakeRoot, Link, Cut.
40. LCT. Link Cut Tree. Оценка Expose и MakeRoot со Splay-деревом.
41. MST за $O(n + m)$. Оценка $\min(m \log n, n^2)$ в худшем.
42. RMQ в offline через алгоритм Тарьяна.

Критерии оценивания и шкала оценки письменного экзамена №4

| Оценка | Критерии выставления оценки |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

ПИСЬМЕННЫЙ ЭКЗАМЕН №5

Письменный экзамен №5 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. Matching. Определения и сложность задач в двудольном и произвольном графе: паросочетание, вершинное покрытие, независимое множество, совершенное паросочетание. Матрица Татта (без док-ва).
2. Matching. Лемма о дополняющем пути.
3. Matching. Алгоритмы поиска паросочетания: обычный, Куна.

4. Matching. Оптимизации алгоритма Куна: не чистить пометки, вообще не чистить пометки, жадная инициализация, быстрое обнуление.
5. Matching. Теорема Кёнига, поиск вершинного покрытия за $O(V + E)$.
6. Matching. Применение Куна для поиска паросочетания в произвольном графе.
7. Matching. Классификация рёбер двудольного графа по принадлежности паросочетанию.
8. Matching. Разбиение вершин орграфа на циклы. Разбиение вершин ациклического орграфа на минимальное число путей.
9. Matching. Теорема Дилворта, поиск максимальной антицепи.
10. Matching. Stable (marriage problem). Алгоритм за $O(E)$.
11. Венгерский алгоритм. Задача о назначениях.
12. Покраски. Рёберная. Теорема Визинга. Связь с паросочетанием. Покраска рёбер двудольного графа.
13. Покраски. Алгоритмы рёберной покраски регулярного двудольного графа за $O(\text{Matching} \cdot \log D)$.
14. Покраски. Алгоритмы рёберной покраски двудольного графа за $O(E^2)$.
15. Покраски. Вершинные. Брукс, практически ценный алгоритм.
16. Потоки. Def: поток, разрез, величина потока, величина разреза, остаточная сеть, циркуляция, прямые и обратные рёбра, декомпозиция потока.
17. Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона.
18. Поиск min разреза по max потоку. Декомпозиция потока за $O(E^2)$, $O(V E)$.
19. k рёберно и вершинно непересекающихся путей в орграфе и неорграфе. Поиск паросочетания, связь контролирующего множества с разрезом.
20. Алгоритм Эдмондса-Карпа. Существование max потока и целочисленного max и потока.
21. Алгоритм масштабирования потока.
22. Алгоритм Диница и его спаривание с масштабированием.
23. Алгоритм Хопкрофта-Карпа и две теоремы Карзанова.
24. $[L, R]$ -циркуляция, $[L, R]$ -flow, $[L, R]$ -max-flow.
25. Глобальный разрез: алгоритм Штор-Вагнера без доказательства. Диниц + link-cut.
26. Глобальный разрез: алгоритм Каргера-Штейна.
27. Mincost k-flow, mincost max flow, mincost circulation, транспортная задача. Формулировки задач. Критерий оптимальности mincost k-flow.
28. Алгоритмы построения mincost k-flow и mincost flow через дополняющие пути в графе без отрицательных циклов.
29. Mincost k-flow. Дейкстра и потенциалы.
30. Графы с отрицательными циклами. Алгоритм Клейна. Полиномиальная версия (без доказательства времени работы).
31. Capacity scaling. Псевдополиномиальный алгоритм для mincost k-flow.
32. Префикс функция. Поиск подстроки в строке и периода строки.
33. Z-функция. Поиск подстроки в строке и периода строки.
34. Алгоритмы Боэра-Мура поиска подстроки в строке.
35. Хеши: полиномиальный хеш, хеш подстроки, Алгоритм Рабина-Карпа поиска подстроки в строке с $O(1)$ доппамяти.
36. Хеши: вероятности. 3 леммы. Каким хешом пользоваться? Обоснование вероятности коллизии в худшем и в среднем.
37. Хеши: антихеш тесты. Строка Туэ-Морса.
38. Хеши: поиск общей подстроки, поиск LCP, построение суффикс-массива за $O(n)$

$\log_2 n$).

39. Палиндромы. Подсчет числа за $O(n \log n)$ хешами и $O(n)$ алгоритмом Манакера. Самый длинный за $O(n)$.

40. Суффмассив: построение за $O(n \log n)$. Поиск подстроки в тексте.

41. Суффмассив: алгоритм Касаи за $O(n)$.

42. Суффмассив: построение за $O(n)$, Каркайнен-Сандерс.

43. Суффмассив: поиск строки в тексте за $O(S + \log T)$.

44. Бор, сжатый бор, хранение. Построение суффдерева за $O(n^2)$ двумя способами. Подстрока в тексте.

45. Автомат Ахо-Корасик. Поиск словарных слов в тексте.

46. Суффдерево: алгоритм Укконена построения за $O(n)$.

47. Преобразование суффдерева \leftrightarrow (суффмассива + LCP) за $O(n)$.

48. Решение задач суффмассивом и суффдеревом: поиск общей подстроки, число различных подстрок.

Критерии оценивания и шкала оценки письменного экзамена №5

| Оценка | Критерии выставления оценки |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопрос не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

7.2.3. Итоговый контроль по дисциплине

ПИСЬМЕННЫЙ ЭКЗАМЕН №6

Письменный экзамен №6 проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа выделяется 2,5 часа.

Примерный перечень вопросов к экзамену:

1. Игры на графах. Ациклический граф. Граф с циклами. Ретроанализ. Длина игры.
2. Игры. Функция Гранди. Прямая сумма игр. Эквивалентность игр. Подсчет через mex.
3. Игры. Сумма игр через XOR. Задачи на Гранди: задача про скамейки, hackenbush.
4. ТЧ. Решето Эратосфена. Версия за $O(n \log \log n)$, время работы, оптимизация в константу раз.
5. ТЧ. Решето Эратосфена. Версия за $O(n)$.
6. ТЧ. Подсчет мультипликативных функций на $[1, n]$ за $O(n)$: φ , число делителей, сумма делителей.
7. ТЧ. Расширенный алгоритм Евклида, способы обращения числа. Время работы.

- Поиск обратных к $1..k$ за $O(k)$.
8. ТЧ. RSA. Алгоритм кодирования, декодирования, сложность вычислений. Взлом в частных случаях.
 9. ТЧ. Дискретный логарифм за $O(\sqrt{p})$. Протокол Диффи—Хеллмана.
 10. ТЧ. Первообразный корень: проверка, поиск (без док-ва вероятности попадания). Применение: корень k -й степени по простому модулю – сведение к делению.
 11. ТЧ. КТО. Использование в длинной арифметики.
 12. Гаусс для полей R , F_p за $O(n^3)$. Решение для треугольной матрицы за $O(n^2)$. Реализация Гаусса для невырожденной квадратной матрицы. Гаусс в F_2 за $O(n^3/w)$.
 13. Реализация Гаусса для произвольной матрицы $m \times n$. Свободные переменные. Базис решений. Разложение вектора в базисе.
 14. Гаусс и погрешность. Матрица Гильберта. Методы борьбы с погрешностью. Простой метод итераций.
 15. Гаусс. Вычисление определителя и обратной матрицы за $O(n^3)$.
 16. Гаусс. Вероятностные задачи на графах – вычисление вероятностей и матожиданий.
 17. Гаусс. Разложение вектора в базисе. Проверка принадлежности точки параллелепипеду. Расстояние от точки до подпространства.
 18. Четыре русских. Умножение матриц над F_2 за $O(n^3/(w \log n))$.
 19. Четыре русских. НОП над константным алфавитом за $O(n^2/\log n)$.
 20. FFT. База: извлечение корня из комплексного числа, единственность интерполяционного многочлена. Схема умножения многочленов за $O(n \log n)$. Интерполяция за $O(n^2)$ (см. практику).
 21. FFT. Рекурсивное вычисление DFT за $O(n \log n)$ (над комплексными числами).
 22. FFT. Нерекурсивная оптимизированная реализация за $O(n \log n)$, обратная битовая запись за $O(n)$.
 23. FFT. Связь обратного DFT и прямого DFT. Вычисление обратного за $O(n \log n)$.
 24. FFT. Целочисленное DFT. Оптимизация: два вещественных DFT в одном комплексном.
Ограничение на коэффициенты для комплексного случая.
 25. FFT. Умножение чисел за $O(n \log n)$, выбор системы счисления. Возведение в степень.
 26. FFT. Поиск с ошибками. Поиск с ошибками и шаблоном. (см. практику)
 27. Длинная арифметика: хранение, линейные операции, деление чисел за $O(n^2)$. Gcd за $O(n^2)$.
 28. Деление чисел и многочленов/чисел за $O(n \log^2 n)$ методом разделяй и властвуй.

Критерии оценивания и шкала оценки письменного экзамена №6

| Оценка | Критерии выставления оценки |
|--------|-----------------------------|
|--------|-----------------------------|

| | |
|--------------------------------|---|
| «Отлично» (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. |
| «Хорошо» (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. |
| «Удовлетворительно» (4-5) | Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не аргументированы. |
| «Неудовлетворительно» (0-3) | Обучающийся не знает значительной части программного материала, допускает существенные ошибки. |

7.3 Порядок формирования оценок по дисциплине

Результирующая оценка по дисциплине рассчитывается следующим образом:

$$O_{рез} = (O_{ум.1} + O_{ум.2} + O_{ум.3} + O_{ум.4} + O_{ум.5} + O_{ум.6}) / 6, \text{ где}$$

$$O_{ум.1} = 0,59 * O_{д/31} + 0,41 * O_{экз1} \text{ (оценка за первый модуль)}$$

$$O_{ум.2} = 0,59 * O_{д/32} + 0,41 * O_{экз2} \text{ (оценка за второй модуль)}$$

$$O_{ум.3} = 0,59 * O_{д/33} + 0,41 * O_{экз3} \text{ (оценка за третий модуль)}$$

$$O_{ум.4} = 0,59 * O_{д/34} + 0,41 * O_{экз4} \text{ (оценка за четвёртый модуль)}$$

$$O_{ум.5} = 0,59 * O_{д/35} + 0,41 * O_{экз5} \text{ (оценка за пятый модуль)}$$

$$O_{ум.6} = 0,59 * O_{д/36} + 0,41 * O_{экз6} \text{ (оценка за шестой модуль)}$$

Действует следующий способ округления накопленной оценки: при значениях от 0,1 до 0,4 оценка округляется в меньшую сторону, от 0,5 до 0,9 – в большую.

На экзамене студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

8. Образовательные технологии

Основными образовательными технологиями являются: интерактивные лекции, работа в группах на практических занятиях.

9. Учебно-методическое и информационное обеспечение дисциплины

9.1 Основная литература

1. Introduction to Algorithms / Cormen, Thomas H. MIT Press. 2009
2. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.:КУРС, НИЦ ИНФРА-М, 2016. - 240 с.

9.2 Дополнительная литература

1. Algorithms Unlocked / Cormen, Thomas H. MIT Press. 2013Verlag, 2008, 728 p. DATA STRUCTURES AND ALGORITHMS USING JAVA™ By: William McAllister. Jones and Bartlett Learning.2009
2. Ахо А. В. Хопкрофт Дж.Э., Ульман Дж. Д. Структуры данных и алгоритмы, уч. пособие, ООО «И.Д. Вильямс», 2010, 398 с.
3. The Algorithm Design Manual; Second Edition / Skiena, Steven S; Wheeler. Springer Verlag London Ltd. 2008
4. Кубенский А.А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++, уч. пособие, СПб.: БХВ-Петербург, 2004, 465 с.

9.3 Программные средства

Практические занятия проводятся в компьютерных классах с выходом в Интернет и доступом к ресурсам электронной библиотеки (<http://library.hse.ru/e-resources/e-resources.htm>) НИУ ВШЭ. Предусматривается наличие у каждого студента рабочего места. Практическая работа ориентирована на использование современных интегрированных инструментальных сред разработки.

Примечание: для самостоятельной работы (установки на персональных компьютерах) студенты могут использовать другие C++ IDE – например, Code::Blocks 15.12 (www.codeblocks.org) / RC1 (по состоянию на 29.12.15), JetBrains CLion 1.2.x (<https://www.jetbrains.com/clion/>).

Используемое программное обеспечение:

1. Microsoft Office Professional 2007-2010 (или более поздние версии),
2. MS Visual Studio 2012 (или более поздние версии).

10. Рекомендации для самостоятельной работы студентов

Самостоятельная работа может рассматриваться как организационная форма обучения – система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных на лекциях теоретических знаний;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;
- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы

и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятиях

11. Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Классы для практических занятий, оснащенные компьютерами с установленной инструментальной средой разработки MS Visual Studio 2012 (или более поздние версии). Для самостоятельной работы необходимо следующее ПО: GNU C++, Oracle Java, Python.

12. Особенности организации обучения для лиц с ограниченными возможностями здоровья

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться следующих варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1) *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2) *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3) *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.