Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет Компьютерных наук
Департамент больших данных и информационного поиска
Базовая кафедра Яндекс

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Науки о данных»
по направлению 01.04.02
«Прикладная математика и информатика»
С.О. Кузнецов

_____
«__» _____ 2015 г.

Программа дисциплины «Algorithms and data structures for search»
для направления 01.04.02 "Прикладная математика и информатика" подготовки магистра
для магистерской программы "Науки о данных"

Автор программы:
Бабенко М.А., к.ф.-м.н. (maxim.babenko@gmail.com)

Одобрена на заседании базовой кафедры Яндекс « » 2015 г.

Заведующий кафедрой М.А. Бабенко

Рекомендована Академическим советом образовательной программы «Науки о данных» « » 2015 г.

Менеджер базовой кафедры Яндекс И.И. Алескерова

Москва, 2015

Пояснительная записка

Автор программы

Бабенко М.А., к.ф.-м.н.

Требования к студентам

Изучение курса «Algorithms and data structures for search» требует базовых знаний по комбинаторике, элементарной теории вероятностей и математическому анализу.

Аннотация

Дисциплина «Algorithms and data structures for search» предназначена для подготовки магистров 01.04.02– Прикладная математика и информатика.

В ходе изучения курса студенты изучат основные методы, приемы и структуры данных, которые используются при создании эффективных алгоритмов и структур данных. В частности, будут изучены основные алгоритмы сортировки и поиска информации; фундаментальные идеи, лежащие в основе данных методов, а также способы их применения на практике; методы оценки сложности алгоритмов.

Программа курса предусматривает лекции (30 часов) и практические занятия (34 часов).

Учебные задачи курса

Цель курса – обучить основным методам и приемам, применяемым при создании эффективных алгоритмов и структур данных.

В результате изучения дисциплины «Algorithms and data structures for search» студенты должны:

- знать основные алгоритмы сортировки и поиска информации;
- понимать фундаментальные идеи, лежащие в основе данных методов, а также способы их применения на практике;
- уметь реализовывать вышеуказанные алгоритмы и оценивать их сложность.

Тематический план дисциплины «Algorithms and data structures for search»

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Сем.и практика	
1	Тема 1.Основные понятия и простейшие алгоритмы	60	8	10	40

2	Тема 2.Структуры данных.	64	10	10	42
3	Тема 3.Графы, автоматы и грамматики	66	12	14	44
	Итого	190	30	34	126

I. Источники информации

Список литературы

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960 с.
2. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. - СПб.: БХВ-Петербург, 2003. – 654 с.
3. Crochemore M., Rytter W. Jewels of Stringology: Text Algorithms. World Scientific Publishing, 2002. – 320 pp.
4. Jurafsky D., Martin J. Speech and Language Processing.PrenticeHall. 1999. – 950 pp.

Дополнительнаялитература

1. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов Издательство: Мир, Москва 1979

II. Формы контроля и структура итоговой оценки

• Текущий контроль: - письменная аудиторная контрольная работа в каждом модуле (60 мин.) и 5 индивидуальных домашних заданий (2+3).

• Промежуточный контроль - зачет в конце третьего модуля;

• Итоговый контроль – письменный экзамен (120 мин.)

Формирование оценки.

Оценка работы студентов на семинарских и практических занятиях, $O_{аудиторная,}$,формируется по десятибалльной шкале и выставляется в рабочую ведомость перед промежуточным и перед итоговым контролем. При формировании оценки учитывается: активность на семинарских занятиях, правильность решения задач на семинаре, результаты письменных тестовых опросов.

Результирующая оценка за текущий контроль в третьем модуле учитывает результаты студента по текущему контролю следующим образом:

$$O_{текущий} = 0.2\ O_{дз1} + 0.2\ O_{дз2} + 0{,}3 \cdot O_{к/р} + 0{,}3 \cdot O_{аудиторная};$$

Результирующая оценка за итоговый контроль в третьем модуле в форме зачета выставляется по следующей формуле, где $O_{зач}$ – оценка за работу непосредственно на зачете:

$$O_{итоговый1} = 0{,}4 \cdot O_{зач} + 0{,}6 \cdot O_{текущий}.$$

Результирующая оценка за текущий контроль в четвертом модуле учитывает результаты студента по текущему контролю следующим образом:

$$O_{текущий} = 0{,}2 O_{дз3} + 0{,}2\ O_{дз4} + 0{,}2\ O_{дз5} + 0{,}2 \cdot O_{к/р} + 0.2\ O_{аудиторная};$$

Результирующая оценка за итоговый контроль в форме экзамена выставляется по следующей формуле, где $O_{экзамен}$ – оценка за работу непосредственно на экзамене:

$$O_{итоговый} = 0{,}4 \cdot O_{экзамен} + 0{,}3 \cdot O_{текущий} + 0{,}3 \cdot O_{итоговый1}.$$

В диплом ставится оценка за итоговый контроль, которая является результирующей оценкой по учебной дисциплине.

Таблица соответствия оценок по десятибалльной и системе зачет/незачет

Оценка по 10-балльной шкале	Оценка по 5-балльной шкале
1	Незачет
2	
3	
4	Зачет
5	
6	
7	
8	
9	
10	

Таблица соответствия оценок по десятибалльной и пятибалльной системе

По десятибалльной шкале	По пятибалльной системе
1 – неудовлетворительно 2 – очень плохо 3 – плохо	неудовлетворительно – 2
4 – удовлетворительно 5 – весьма удовлетворительно	удовлетворительно – 3
6 – хорошо 7 – очень хорошо	хорошо – 4
8 – почти отлично 9 – отлично 10 - блестяще	отлично – 5

III. Программа дисциплины «Algorithms and data structures for search»

Тема 1. Основные понятия и простейшие алгоритмы

Вычислительные модели. Простейшие структуры данных.Классификация задач по трудности решения. Основные ресурсы: память и время. RAM-машина. Анализ учетных стоимостей операций. Структуры данных: понятие об интерфейсе и реализации. Массивы переменного размера. Связные списки. Стеки, очереди и деки.

Задача сортировки набора ключей. Стабильные и нестабильные сортировки. Нижняя оценка числа сравнений в модели разрешающих деревьев. Процедура Partition разделения массива на две части. Алгоритм Quick-Sort. Сложность в среднем и худшем случае. Способы выбора разделителя. Элиминация хвостовой рекурсии. Слияние двух упорядоченных списков. Алгоритм Merge-Sort. Оценка сложности. Сортировка слиянием без использования дополнительной памяти.

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960 с.
2. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. - СПб.: БХВ-Петербург, 2003. – 654 с.
3. Crochemore M., Rytter W. Jewels of Stringology: Text Algorithms. World Scientific Publishing, 2002. – 320 pp.
4. Jurafsky D., Martin J. Speech and Language Processing.PrenticeHall. 1999. – 950 pp.

Дополнительнаялитература

1. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов Издательство: Мир, Москва 1979

Тема 2.Структуры данных

Деревья поиска.Определение дерева поиска. Нахождение минимума и максимума. Нахождение последователя и предшественника. Вставка и удаление элементов. Высота дерева, понятие о сбалансированности. Вращения. Деревья поиска с большим коэффициентом ветвления (2,3-деревья, 2,3,4-деревья, B-деревья). Красно черные деревья: определение и основные свойства. Связь между красно-черными и 2,3,4-деревьями. AA-деревья. Связь между AA-деревьями и 2,3-деревьями. Реализация операций вставки и удаления для AA-деревьев.

Порядковые статистики.Понятие о k-й порядковой статистике. Нахождение с помощью модификации алгоритма Merge-Sort. Нахождение в режиме online с помощью приоритетной очереди. Нахождение с помощью рандомизированной модификации алгоритма Quick-Sort. Линейность матожидания времени работы. Приближенные медианы. Выбор k-й порядковой статистики за линейное в худшем случае время. Сравнение изученных подходов.

Хеширование.Понятие о хеш-функции. Открытая адресация. Коллизии. Разрешение коллизий методом последовательных проб. Разрешение коллизий методом двойного хеширования. Разрешение коллизий методом цепочек. Примеры хеш-функций. Bloomfilter. Реализация на основе массива бит и на основе массива счетчиков.

Деревья со свойствами кучи. Почти полные бинарные деревья: нумерация вершин, навигация. Двоичная куча. Операция просеивания вниз и вверх. Реализация операций вставки, удаления и поиска минимума. Преобразование произвольного массива ключей в кучу, линейность времени работы. Алгоритм сортировки Heap-Sort. Lefist-кучи. Ранги вершин. Логарифмическая оценка на длину правого пути в leftist-куче. Слияние leftist-куч.

Реализация операций удаления минимума и вставки через операцию слияния. Skew-кучи. Логарифмическая учѐтная оценка для сложности операции объединения. Immutable-структуры данных. Хранение истории изменений. Декартовы деревья. Единственность декартова дерева для заданного набора пар ключей и приоритетов. Дучи (treaps). Логарифмическая оценка матожидания высоты дучи. Операции слияния и разделения для дуч. Операции вставки и удаления элементов для дуч.

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960 с.
2. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. - СПб.: БХВ-Петербург, 2003. – 654 с.
3. Crochemore M., Rytter W. Jewels of Stringology: Text Algorithms. World Scientific Publishing, 2002. – 320 pp.
4. Jurafsky D., Martin J. Speech and Language Processing.PrenticeHall. 1999. – 950 pp.

Дополнительнаялитература

1. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов Издательство: Мир, Москва 1979
2. Czech Z., Havas G. An Optimal Algorithm for Generating Minimal Perfect Hash Functions.InformationProcessingLetters. Vol. 42. 1992. Pp. 257-264.

Тема 3. Графы, автоматы и грамматики

Алфавит и строки в нем: основные определения и обозначения. Алфавитнозависимые и алфавитнонезависимые оценки сложности алгоритмов. LCP- и Z-функции. Блоки строки относительно Z-функции. Алгоритм вычисления Z-функции за линейное время. Варианты задачи поиска подстроки в строке. Использование Z-функции для предобработки и поиска подстроки. Понятие об алгоритме поиска реального времени. Алгоритм Кнута-Морриса-Пратта, префикс-функция. Алгоритм построения префикс-функции. Линейность времени его работы. Задача множественного поиска подстрок в строке. Использование алгоритма Рабина-Карпа. Бор для множества слов. Способы хранения: открытая адресация, списки дуг, сбалансированные деревья дуг, хеш-таблицы. Функция откатов на боре. Построение за линейное время.

Суффиксные деревья.Сжатые и несжатые суффиксные деревья. Линейность размера сжатого суффиксного дерева. Явные и неявные положения. Суффиксные ссылки. Общая схема алгоритма Укконена. Общий вид преобразований, совершаемых алгоритмом на каждой фазе. Суффиксные положения, их классификация. Прием неявного продления листовых дуг. Прием скачка по счетчику. Вычисление стартового положения для алгоритма на следующей фазе по конечному положению на предыдущей. Линейность времени работы.

Задачи LCA (least common ancestor) связьсфункцией LCP. Эйлеров обход дерева. Сведение задачи LCA к задаче RMQ (rangeminimumquery). Таблица ответов для задачи RMQ.

Спарсификация таблицы. Решение задачи RMQ со сложностью O(n log n) для преобработки и O(1) для ответа на запрос. Trade-off между временем предобработки и временем ответа на запрос. Задача ±1-RMQ. Связь с задачей LCA. Решение задачи ±1-RMQ со сложностью O(n) для преобработки и O(1) для ответа на запрос.

Конечныеавтоматы (finite state automata). Варианты определений: детерминированные и недетерминированные автоматы. Эпсилон-переходы. Эпсилон-замыкание. Детерминизация. Экспоненциальный рост числа состояний при детерминизиации. Эквивалентность автоматов. Минимальные автоматы. Квадратичный алгоритм построения минимального автомата. Оптимизация процесса построения DFA по NFA, обход пространства состояний. Сортировка подсчетом. Поразрядная сортировка. Использование поразрядной сортировки для решения задачи минимизации автомата за квадратичное время. Изоморфизм автоматов. Теорема о тождественности понятий измоморфизма и эквивалентности для минимальных автоматов. Поиск изоморфизма для пары минимальных автоматов. Преобразователи с конечным числом состояний (finitestatetransducers). Регулярные выражения и регулярные языки. Совпадение классов автоматных и регулярных языков (теорема Клини). Лемма о накачке для регулярных языков. Примеры нерегулярных языков.

Использование конечных автоматов для лексического разбора текста. Основные понятия теории морфологического анализа: фонемы, морфемы, слова, корни, аффиксы, флексии, основы. Словообразование и словоизменение. Лексемы, парадигмы и леммы. Примеры лингвистических явлений, связанных с морфологией: дефекты парадигм, супплетивизм, омонимия. Глубинная (лексическая), промежуточная и поверхностная структура слов при морфологическом анализе. Реализация лексикона, морфотактики и орфографических правил при помощи конечных преобразователей. Операции над преобразователями: конкатенация и обращение. Вычисление образа строки под действием преобразователя.

Оптимальные выравнивания последовательностей. Редакторское расстояние (метрика Левенштейна). Связь с задачей об экстремальных путях в ациклическом графе. Восстановление оптимального решения по таблице динамического программирования. Memoization. Скрытые марковские модели. Скрытые и наблюдаемые состояния. Алгоритмы forward-backward, Viterbi.

Контекстно-свободные грамматики и задаваемые ими языки. Лемма о накачке для контекстно-свободных языков. Нормальная форма Хомского. Приведение грамматики к нормальной форме Хомского. Деревья разбора. Однозначные и неоднозначные грамматики. Алгоритм CYK (Cocke-Younger-Kasami). Вычисление множества Nullable-нетерминалов. Алгоритм Earley. Левосторонние и правосторонние выводы. Табличные LL(1)-парсеры. Вычисление First-множеств. Вычисление Follow-множеств. Заполнение таблиц LL(1)-парсера. Пример: грамматика арифметических выражений. Понятие о приоритете и ассоциативности операций. Метод рекурсивного спуска.

Графы: основные определения, обозначения и способы хранения. Связность в ориентированных графах. Отношения достижимости, контрдостижимости и взаимной достижимости вершин. Обход в ширину и его использование для нахождения кратчайших путей при единичных длинах дуг. Алгоритм Dial для случая единичных и нулевых длин дуг. Обход в глубину и его основные свойства. Дерево обхода в глубину. Классификация дуг графа относительно дерева обхода в глубину (дуги дерева, обратные, прямые, перекрестные). Моменты времени начала и конца обработки вершин. Поиск циклов в ориентированных

графах. Топологический порядок. Построение топологического порядка графа с помощью обхода в глубину. Компоненты сильной связности, конденсация. Поиск сильно связных компонент с помощью обхода в глубину. Топологическая сортировка конденсации.

Задача о кратчайших путях и ее варианты (APSPP, SSSPP). Функции длин путей (аддитивная, максимум длин дуг). Алгоритм Форда-Беллмана. Алгоритм Флойда. Алгоритм Дейкстры.

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960 с.
2. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. - СПб.: БХВ-Петербург, 2003. – 654 с.
3. Crochemore M., Rytter W. Jewels of Stringology: Text Algorithms. World Scientific Publishing, 2002. – 320 pp.
4. Jurafsky D., Martin J. Speech and Language Processing.PrenticeHall. 1999. – 950 pp.

Дополнительнаялитература

1. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов Издательство: Мир, Москва 1979

IV. Списоклекций

Lecture 1. Breadth first search. Depth first search (part 1)

Graphs: basic notation, in-memory representation. Computing shortest paths with breadth first search. Depth first search and its basic properties.

Lecture 2. Depth first search (part 2)

Euler tours. Computing Euler tours with DFS. DFS tree and types of edges. Undirected graphs have no cross edges. Computing articulation points in linear time with DFS.

Lecture 3. Depth first search (part 3)

Bidirectional reachability. Strongly connected components. Graph condensation and its acyclicity. Computing strongly connected components via DFS. 1- and 2-cuts. Vector space spanned by graph edges and its dimension. Cycle subspace, its dimension and basis. Cut subspace. Orthogonal decomposition. Sampling elements in cycle space. Finding 2-cuts via randomized fingerprints.

Lecture 4. Shortest paths (part 1)

Length functions and shortest paths. Distance labels and their relaxation. Ford--Bellman algorithm. Floyd algorithm. Dijkstra algorithm.

Lecture 5. Shortest paths (part 2)

Complexity of Dijkstra algorithm. Power of data structures: binary heaps, k-ary heaps. Bidirected Dijkstra algorithm. Dual approach to shortest paths: potentials. Feasible potentials and

conservative edge lengths. Johnson algorithm for computing all-pairs shortest paths for arbitrary edge lengths. Speeding up search with landmarks. ALT algorithm.

Lecture 6. Minimum spanning trees

Minimum spanning tree problem. Good edge sets. Extending good edge sets by adding minimum cut edges. Kruskal algorithm, Prim algorithm, Boruvka algorithm and their complexities.

Lecture 7. Minimum cuts. Substring matching (part 1)

Minimum s-t cut and minimum global cut problems. Solving minimum global cut problem via a sequence of s-t problems. Contractions. Stoer-Wagner algorithm. Strings: basic notation and definitions. Substring matching problem. Naive algorithm. Borders and prefix function. Computing prefix function in linear time. Knuth-Morris-Pratt algorithm.

Lecture 8. Substring matching (part 2)

Z-function. Substring matching via Z-function. Constructing Z-function in linear time. Reducing space from O(P+T) to O(P). Computing 1-approximate substring matches via Z-function. Multiple pattern matching. Word trie: definition, in-memory representation and construction. Failure function. Aho-Corasick algorithm.

Lecture 9. Substring matching (part 3)

Matching with ?- and *-wildcards. Matching with *-wildcards via Aho-Corasick. Matching with ?-wildcards via convolutions. Reducing convolutions to polynomial multiplication. Further improvements: making alphabet binary, double cover trick.

Lecture 10. Substring matching (part 4). Suffix trees (part 1)

FFT: definition and recursive implementation. Inverse FFT. Suffix trie and suffix tree. Space bounds. Explicit and implicit locations. Suffix links.

Lecture 11. Suffix trees (part 2). Suffix arrays (part 1)

Outline of Ukkonen algorithm. Iterations and steps. Step types. Evolution of step types. Eliminating Type 1 steps via implicit labels at leaf edges. Eliminating Type 3 steps via early exit. Bounding the number of Type 2 steps. Finding locations for Type 2 steps via suffix links. Skip-count trick for computing suffix links. Following a suffix link does not decrease the current depth by most than 1. Bounding the running time of Ukkonen algorithm. Suffix arrays. Finding substring matches via suffix array and binary search. Karp-Miller-Rosenberg labeling algorithm for computing suffix arrays in O(n log n) time.

Lecture 12. Suffix arrays (part 2).

Speeding up substring matching with LCP values. Karkainen-Sanders algorithm for linear-time suffix array construction. LCAs in suffix tree are related to LCPs. LCP array: definition and

basic properties. Computing LCP array from suffix array in linear time (Arikawa-Arimura-Lee-Kasai-Park algorithm).

Lecture 13. Longest common substrings. Approximate substring matching.

Solving longest common substring problem via suffix trees and suffix arrays. Approximate substring matching with edit distance. Alignment graph and dynamic programming approach. Landau-Vishkin algorithm. Reachable sets and their boundaries. Initializing and updating boundary set. LCPs to the rescue.

V. Методические указания студентам

Самостоятельная работа студента предусматривает выполнение теоретических заданий, направленных на овладение техникой построения алгоритмов поиска и сортировки, а также практических заданий по программной реализации этих алгоритмов.

Автор программы: _ / <Бабенко М.А.> /