

Алгоритмы и структуры данных.

Часть 2

ДЕНИС СЕРГЕЕВИЧ КРАХМАЛЁВ

АСПИРАНТ МФТИ
JAVA-РАЗРАБОТЧИК

СПИКЕР КУРСА "ПРОФЕССИЯ РАЗРАБОТЧИК НА C++" ОНЛАЙН-ШКОЛЫ SKILLBOX

9 ЯНВАРЯ 2023 Г.

Цель данного курса – изучить классический набор алгоритмов и структур данных, используемых в программировании, охватив все области алгоритмических задач – от поиска по массиву до работы со строковыми данными и геометрией. В ходе курса студенты не только получают теоретические знания в теории алгоритмов, но и разовьют алгоритмическое мышление, необходимое для эффективного решения программных задач.

Курс

Обходы графа.

Ориентированный граф, неориентированный граф, псевдограф.

Обход в глубину. Цвета вершин. Времена входа и выхода. Лемма о белых путях.

Проверка связности неориентированного графа.

Поиск цикла в неориентированном и ориентированном графе.

Топологическая сортировка.

Связность в неор. графе, компоненты связности. Слабая и сильная связность в ор. графе. Компоненты слабой, сильной связности.

Нахождение компонент сильной связности. Алгоритм Косарайю с доказательством корректности. Алгоритм Тарьяна с доказательством корректности.

Компоненты реберной двусвязности. Мосты. Поиск мостов.

Компоненты вершинной двусвязности. Точки сочленения. Поиск точек сочленения.

Волновой алгоритм. Обход в ширину (применение очереди в волновом алгоритме).

Критерий существования Эйлера пути и цикла в ориентированном и неориентированном графе. Поиск эйлера пути и цикла.

Кратчайшие пути во взвешенном графе

Алгоритм Дейкстры. Доказательство корректности. Оценка времени работы. Дерево кратчайших путей.

Алгоритм Флойда. Доказательство. Восстановление пути.

Нахождение цикла отрицательного веса.

Алгоритм Форда-Беллмана.

Хранение в матрице: Dvk равно длине кратчайшего пути до вершины v за ровно k ребер (не более k ребер). Доказательство корректности. Оценка времени работы.

Восстановление пути.

Детектирование цикла отрицательного веса. Поиск самого цикла.

Нахождение кратчайших путей с учетом циклов отрицательного веса.

Алгоритм Йена

Алгоритм Левита, поиск цикла минимального веса.

Алгоритм A^* . Условие монотонности на эвристику. Допустимость эвристики. Примеры эвристик.

Остовное дерево. Построение с помощью обхода в глубину и в ширину.

Определение минимального остовного дерева.

Теорема о разрезе. Доказательство.

Алгоритм Прима. Аналогия с алгоритмом Дейкстры.

Алгоритм Крускала. Доказательство. Оценка времени работы.

Система непересекающихся множеств. Эвристика ранга с доказательством оценки времени работы.

Эвристика сжатия пути без доказательства.

Алгоритм Борувки. Доказательство. Оценка времени работы.

Приближение решения задачи коммивояжера с помощью минимального остовного дерева.

Определение сети. Определение потока.

Физический смысл. Аналогия с законами Кирхгофа.

Определение разреза. Понятия потока через разрез.

Доказательство факта, что поток через любой разрез одинаковый.

Понятие остаточной сети. Понятие дополняющего пути.

Необходимость отсутствия дополняющего пути для максимальности потока.

Теорема Форда-Фалкерсона.

Алгоритм Форда-Фалкерсона. Поиск минимального разреза.

Пример целочисленной сети, в котором алгоритм работает долго.

Алгоритм Эдмондса-Карпа.

Доказательство, что кратчайшее расстояние в остаточной сети не уменьшается.

Общая оценка времени работы алгоритма Эдмондса-Карпа.

Слоистая сеть. Алгоритм Диница. Оценка времени работы.

Паросочетания. Паросочетание в двудольном графе. Максимальное паросочетание.

Чередующие пути. Увеличивающие пути. Теорема Берга.

Сведение поиска максимального паросочетания к поиску максимального потока.

Алгоритм Куна.

Поиск строк.

Понятие префикс, суффикс, подстрока. Понятие собственных префикса и суффикса.

Постановка задачи поиска подстроки в строке. Тривиальный алгоритм поиска подстроки в строке.

Префикс-функция. Тривиальный алгоритм нахождения.

Линейный алгоритм нахождения. Доказательство времени работы

Подсчет префикс-функции для строки $q\$t$, где q — образец, а t — текст.

Алгоритм Кнута-Морриса-Пратта. Поточная обработка текста без хранения префикс-функции для всей строки $q\$t$.

Z-функция. Тривиальный алгоритм нахождения.

Линейный поиск Z-функции. Доказательство времени работы

Применение для поиска подстроки в строке (КМП-2). Хранение Z-функции только для образца, а не для всей строки $q\$t$.

Построение строки по z-функции

Построение Префикс-функции строки при известной z-функции

Структура данных Бор. Построение, оценка времени построения и объема памяти.

Алгоритм Ахо-Корасик. Суффиксная ссылка. Построение бора. Построение суффиксной ссылки. Оценка времени работы.

Ситуации, когда один образец является суффиксом другого. «Длинные» суффиксные ссылки, то есть ссылки, идущие в следующую терминальную вершину, которая является суффиксом текущей.

Построение автомата переходов. Построение переходов по буквам с учетом перемещения по суффиксным ссылкам. Кэширование переходов.

Бор с суффиксными ссылками для одного шаблона. Аналогия с префикс-функцией. Суффиксные структуры данных.

Суффиксный массив. Построение за $O(n^2 \log n)$.

Поиск подстроки в тексте с использованием суффиксного массива.

Построение суффиксного массива за $O(n \log n)$ с помощью удвоения префикса, по которому происходит цифровая сортировка.

Алгоритм Касаи. Доказательство времени работы.

Суффиксное дерево.

Сжатое суффиксное дерево. Хранение сжатого суффиксного дерева. Тривиальное построение сжатого суффиксного дерева.

Линейность числа вершин и ребер.

Обновление дерева при добавлении одного символа в конец строки. Два случая: создание нового листа и проход вдоль ребра. Эвристика листа. Добавление бесконечного числа символов на ребро при добавлении листа.

Суффиксная ссылка. Инвариант: для всех внутренних вершин вычислена суффиксная ссылка. Доказательство факта, что суффиксная ссылка ведет всегда в вершину.

Переход к суффиксу меньшего размера и подсчет суффиксной ссылки для вновь созданной вершины. Быстрый спуск, обоснование его допустимости.

Алгоритм Укконена. Потенциалы для доказательства времени работы. Потенциал по длине слова, соответствующего вершине. Потенциал по количеству промежуточных вершин от корня.

Доказательство асимптотики

Вычислительная геометрия.

Введение. Точка, вектор, отрезок. Скалярное произведение, векторное произведение. Прямая, плоскость.

Выпуклая оболочка 2D.

Алгоритм Джарвиса. Алгоритм Грэхема.

Метод в 2D «Разделяй и властвуй» за $O(n \log n)$.

Упражнение: Чан в 2D

Выпуклая оболочка 3D. Заворачивание подарка за $O(n^2)$.

Метод в 3D «Разделяй и властвуй» за $O(n \log n)$

Сумма Минковского двух выпуклых многоугольников за $O(m + n)$.

Сканирующая прямая.

Проверка факта пересечения какой-либо пары отрезков из множества за $O(n \log n)$.

Триангуляция Делоне.

Сведение к поиску выпуклой оболочки в 3D.

ЕМОД - Евклидово минимальное остовное дерево. Достаточность использования ребер триангуляции Делоне. $O(V \log V)$.

Диаграмма Вороного.

Эквивалентность триангуляции Делоне.

Построение диаграммы Вороного по схеме: Выпуклая оболочка 3D → Триангуляция Делоне → Диаграмма Вороного. $O(n \log n)$.

Пререквизиты к слушателям

Студент должен владеть материалом первой части курса, в частности темами «Базовые структуры данных» «Сортировки» и «Асимптотические оценки». Студент должен свободно работать с данными в объектно-ориентированном стиле на одном из языков программирования (в частности уметь обращаться с массивами, строками, функциями (в т.ч. рекурсивными), структурами и классами). Желательно иметь базовые представления об объектах линейной алгебры.

Список литературы

Кормен Томас Х., Лейзерсон Чарльз И., Ривест Рональд Л., Штайн Клиффорд “Алгоритмы. Построение и анализ”
Роберт Седжвик “Algorithms in C++”
Дональд Кнут “The Art of Computer Programming”
Бхаргава Адитья “Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих”
Bernhard von Stenge “Game Theory Basics”
Муртаф Б “Современное линейное программирование”

Дополнительная информация

Школа точных наук

Контакты для обратной связи
