Программа “Алгоритмы и структуры данных” 3 семестр 2019-2020

# Тема 1. Поиск строк.

3 лекции

* 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

ю.

* 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.

# 

# Тема 2. Суффиксные структуры данных.

3 лекции

* **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

# Тема 3. Вычислительная геометрия.

5 лекций

* 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).
* Выпуклая оболочка **3D**. Заворачивание подарка за O(n^2).
* Метод в 3D «Разделяй и властвуй» за O(n log n)
* Алгоритм Чана
* **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).
* Алгоритм Форчуна.

# Тема 4. Длинная арифметика. Преобразование Фурье. Комбинаторные игры.

3 лекции

* **Длинная арифметика**.
* Умножение. Алгоритм Карацубы.
* Деление.
* **Преобразование Фурье**. Дискретное преобразование Фурье. Быстрое преобразование Фурье.
* **Математические игры**.
* Игра с камнями. Есть N камней, игрок может брать от 1 до K камней. Побеждает игрок, взявший последний камень.
* Модификация: проигрывает игрок, который взял последний камень.
* Игра в монеты за круглым столом. Игроки по очереди кладут круглые монеты на круглый стол так, чтобы они не пересекались. Игрок, который не может сделать ход, проигрывает. **Метод симметричной стратегии**.
* **Игры на графе**. Выигрышные и проигрышные вершины.
* **Классификация игр**. Нормальные и ненормальные игры. Нормальной называют игру, в которой листья проигрышные. Сведение ненормальной игры к нормальной. Справедливые и несправедливые игры. Справедливые, когда каждый игрок из одной позиции может делать такие же ходы, как и противник. Пример не справедливой игры: шахматы, шашки. Случайные/детерминированные.
* Выигрышная и проигрышная стратегия. Определение оптимальной стратегии с учетом количества ходов. Поиск оптимальной стратегии в ациклических графах. Min-Max.
* **Ретро-анализ**. Нахождение хода в оптимальной стратегии во время ретро-анализа
* **Минимакс.** Оценка состояния игры эвристикой.
* **Альфа-бета отсечение**