Алгоритмы и структуры данных

Экзамен, 2-й семестр

1 Медиана и порядковые статистики. Алгоритм с линейным временем работы для медианы

2 Сортировка массива: пузырьковая, mergesort, quicksort. Алгоритмы и оценки сложности.

3 Списки: односвзяный и двусвязный

4 Бинарные деревья поиска. Вставка, удаление, оценки сложности.

5 Графы. Способы их представления в памяти компьютера. Матрицы смежности, матрицы инцедентности, списки связности, представление на двух массивах (CSR).

6 Обходы графов. Обход в ширину, обход в глубину.

7 Алгоритмы поиска кратчайших путей. Алгоритм Дейкстры, алгоритм Беллмана-Форда.

8 Остовные деревья. Алгоритмы Прима, Краскала, Борувки.

9 Эвристики для поиска кратчайших путей, алгоритм A*.

10 Потоки в сетях. Максимальный поток и минимальный разрез.]

11 Хэш-функции. Коллизии. Хеш-таблицы. Хэширование. Фильтр Блюма.

12 Предикат поворота. Задача пересечения двух отрезков

13 Выпуклые оболочки, алгоритма Джарвиса, Грэхема и QuickHull.

14 kD- деревья. Окто- и квадро-деревья.

15 Многочлены. Метод Горнера. Умножение Карацубы.

16 Основная теорема для рекуррентных соотношений. Схема доказательства.

Теорема

Пусть $T(n) = a \cdot T(\lceil n/b \rceil) + O(n^d)$. Тогда

$$T(n) = \begin{cases} O(n^d), & d > \log_b a \\ O(n^d \log n), & d = \log_b a \\ O(n^{\log_b a}), & d < \log_b a \end{cases},$$

где $a \geq 1$ — количество частей, на которые мы дробим задачу, $b > 1$ — во сколько раз легче становится решить задачу, d — степень сложности входных данных.

Схема доказательства:

Рассмотрим дерево рекурсии данного соотношения. Всего в нем будет $\log_b n$ уровней. На каждом таком уровне, количество детей в дереве будет умножаться на a на уровне i будет a^i потомков. Также известно, что каждый ребенок на уровне i размера $\frac{n}{b^i}$. Ребенок размера $\frac{n}{b^i}$ требует $O\left(\left(\frac{n}{b^i}\right)^d\right)$ дополнительных затрат, поэтому общее количество совершенных действий на уровне i:

$$O\left(a^i \cdot \left(\frac{n}{b^i}\right)^d\right) = O\left(n^d \cdot \left(\frac{a^i}{b^{id}}\right)\right) = O\left(n^d \cdot \left(\frac{a}{b^d}\right)^i\right)$$

Решение разбивается на три случая: когда $\frac{a}{b^d}$ больше 1, равна 1 или меньше 1. Переход между этими случаями осуществляется при

$$\frac{a}{b^d} = 1 \Leftrightarrow a = b^d \Leftrightarrow \log_b a = d \cdot \log_b b \Leftrightarrow \log_b a = d$$

Распишем всю работу в течение рекурсивного спуска:

$$T(n) = \sum_{i=0}^{\log_b n} O\left(n^d \left(\frac{a}{b^d}\right)^i\right) + O(1) = O\left(n^d \cdot \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i\right)$$

Отсюда получаем:

1. $d > \log_b a \Rightarrow T(n) = O(n^d)$ (так как $\left(\frac{a}{b^d}\right)^i$ — бесконечно убывающая геометрическая прогрессия)

2. $d = \log_b a \Rightarrow T(n) = O\left(n^d \cdot \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i\right) =$

   $= O\left(n^d \cdot \sum_{i=0}^{\log_b n} (1)^i\right) = O(n^d + n^d \cdot \log_b n) = O(n^d \cdot \log_b n)$

3. $d < \log_b a \Rightarrow T(n) = O\left(n^d \cdot \sum_{i=0}^{\log_b n} \left(\frac{a}{b^d}\right)^i\right) = O\left(n^d \cdot \left(\frac{a}{b^d}\right)^{\log_b n}\right)$, но

$$n^d \cdot \left(\frac{a}{b^d}\right)^{\log_b n} = n^d \cdot \left(\frac{a^{\log_b n}}{b^{d \log_b n}}\right) = n^d \cdot \left(\frac{n^{\log_b a}}{n^d}\right) = n^{\log_b a} \Rightarrow T(n) = O(n^{\log_b a})$$

17 АВЛ-деревья. Повороты, балансировка.

18 Красно-черные деревья. Балансировка (схема).

19 Бинарные кучи. Реализация с указателями и на массиве. Добавление и удаление элемента в бинарную кучу.

20 Очереди с приоритетами. Наивная реализация, реализация на бинарной куче.

21 Потоки в сетях. Задача о максимальном потоке. Алгоритм Форда – Фалкерсона.

22 Амортизационный анализ: групповой анализ, банковский метод. Амортизационный анализ для бинарного счетчика

23 Амортизационный анализ: метод потенциалов для динамического массива.

24 Алгоритм Рабина-Карпа, алгоритм Кнута-Морриса-Пратта . Структура данных «бор», алгоритм Ахо-Корасик.

25 Алгоритм Бойера-Мура. Эвристики стоп-символа и хорошего суффикса. [7]

26 Расстояние Левенштейна, алгоритм Вагнера-Фишера.

27 Сканирующая прямая. Алгоритм Бентли-Оттоманна для поиска пересечения отрезков

28 Диаграммы Вороного. Алгоритм Форчуна.

29 Триангуляция Делоне, связь с диаграммами Вороного. Алгоритм построения

30 Сумма Минковского. Задача планирования движения робота в среде с препятствиями. Граф видимости.

31 Отсечение невидимых поверхностей. Z-буфер и алгоритм художника.