Список вопросов по темам на зачёт для параллели А

Где-то около Менделеево, август 2009

Структуры данных

1. (3) Определение дерева отрезков. Применение для различных функций. Реализация основных запросов снизу и сверху.
2. (3) Реализация на дереве отрезков запросов `a[i..j] += x` и `a[i..j] = x` сверху (запрос `get` тоже должен быть сверху).
3. (3) Дерево отрезков без сжатия координат ($\mathrm{O}(N \log M)$,$\mathrm{O}(\log M)$). Случаи целых и вещественных чисел.
4. (4) Корневая эвристика. Определение. Сравнение с деревом отрезков. Запросы: `a[i] = x`, `get(i..j)`. Запросы на отрезках `a[i..j] += x`, `a[i..j] = x`. Операции `del(i)`, `insert(i)`, `invert(l..r)`. Переменная длина куска $[\sqrt{N} \ldots 2\sqrt{N}]$ — операции `split` и `merge` для куска.
5. (4) Задача о поиске K-й порядковой статистики. За время (предподсчёт, время на запрос) ($\mathrm{O}(1)$, $\mathrm{O}(n)$), ($\mathrm{O}(n \log n)$, $\mathrm{O}(\log^3 n)$), ($\mathrm{O}(n^2)$, $\mathrm{O}(\log^2 n)$), ($\mathrm{O}(n^3)$, $\mathrm{O}(1)$).
6. (3) Дерево Фенвика. Определение. Запросы `add` и `get`. Доказательство.
7. (4) Использование Дерева Фенвика для функции минимума. Многомерное дерево фенвика. Формула включения-исключения с доказательством (многомерный случай).
8. (4) 2D-дерево отрезков для матрицы $N \times N$. Запрос за $\mathrm{O}(\log^2 n)$.
9. (5) 2D-дерево отрезков для N точек. $\mathrm{O}(n \log n)$ памяти, запрос за $\mathrm{O}(\log^2 n)$. Проблемы с модификацией на прямоугольнике. Запросы `paint(x1, y1, x2, y2)` и `color(x, y)` за $\mathrm{O}(\log^2 n)$.
10. (4) Динамическое 2D-дерево отрезков без сжатия координат $\mathrm{O}(n \cdot \log^2 M)$, $\mathrm{O}(\log^2 M)$.
11. (4) Квадродерево. Определение. Построение расчетверением или раздвоением на каждом уровне. Запрос за $\mathrm{O}(n)$. Доказательство временной оценки.
12. (3) Декартово дерево (treap). Определение. Обоснование логарифмической высоты. Операции `add`, `split`. Операции `del`, `merge`. Реализация `add` и `del` через `split` и `merge`.
13. (4) Применение декартового дерева. `size[v]`. Операции `getPos(x)`, `getX(pos)`. Операции `min(i..j)`, `max(i..j)`, `sum(i..j)`. Операция `invert(i..j)`. Динамическая K-я порядковая статистика на отрезке.
14. (4) Декартово дерево по неявному ключу. Определение. Создание. Задачи, которые можно решать с помощью Декартова дерева по неявному ключу. Операции с массивом: `insert`, `delete`, `split`, `merge`, `invert(i..j)`, `sum(i..j)`, `min(i..j)`.
15. (3) Хеш (hash). Хеш-таблица. Полиномиальный хеш. Отсутствие коллизий. Решение задачи: максимальная по длине общая подстрока двух строк за $\mathrm{O}(N \log N)$.

LCA, RMQ, RSQ, CHM

1. (3) Система непересекающихся множеств. Реализация за время $\sum$ `join` $= \mathrm{O}(n \log n)$, `get` $= \mathrm{O}(1)$.
2. (4) Реализация CHM за время `join` $= \mathrm{O}(1)$, $\sum$ `get` $= \mathrm{O}(\log^* n)$. Доказательство оценки $\mathrm{O}(\log^* n)$ («крутые ребра»).
3. (3) LCA offline за $\mathrm{O}(n)$ (с CHM).
4. (3) LCA за ($\mathrm{O}(n)$, $\mathrm{O}(\log n)$). Отношение «? a предок b» за ($\mathrm{O}(n)$, $\mathrm{O}(1)$). Сведение RMQ к LCA за $\mathrm{O}(n)$. Сведение LCA к RMQ ± 1 за $\mathrm{O}(n)$.
5. (5) RMQ за ($\mathrm{O}(n \log n)$, $\mathrm{O}(1)$), Решение задача RMQ ± 1 алгоритмом 4-х русских ($\mathrm{O}(N)$, $\mathrm{O}(1)$).

6. (?) Решение задача RMQ за ($O(n)$, $O(\log\log n)$) и за ($O(n\log^* n)$, $O(\log^* n)$). Решение задачи про минимум на движущемся окошке за $O(n+m)$ (Запросы: `L++`, `R++`, `min(L, R)`).

7. (4) Использование идеи 4-х русских для поиска наибольшей общей подпоследовательности двух строк над алфавитом {a,b} за $O(n^2/\log^2 n)$.

8. (4) LCA за ($O(n\log n)$, $O(\log n)$) — метод двоичных подъёмов. Решение задачи про подвешивание деревьев друг к другу и LCA за $O(n\log n)$ в offline и online.

9. (4) Функции на путях дерева. Веса не меняются. Сумма на пути ($O(n)$, LCA+$O(1)$). Минимум на пути ($O(n\log n)$, LCA + $O(\log n)$). Минимум на пути offline ($O(n)$, LCA + $O(\log n)$).

10. (4) Функции на путях дерева. Веса меняются. Покрытие дерева деревьями отрезков (предподсчет за $O(n)$) с запросом для пути минимума, максимума, суммы за $O(\log^2 n)$. Сумма на пути ($O(n)$, LCA + RMQ).

11. (4) Минимум на пути online, сведение к 2D дереву отрезков. Использование offline решения минимума в прямоугольнике для решения минимума на пути offline за ($O(n)$, LCA + $O(\log n)$).

Строки

1. (3)Определения, связанные с суффиксным массивом. Циклические сдвиги и суффиксы, получение одного из другого и наоборот. Построение за $O(n^2)$ цифровой сортировкой.

2. (4) Построение суффиксного массива за $O(n\log n)$.

3. (4) Построение LCP за $O(n)$ — алгоритм Касаи. Проблемы алгоритма Касаи в случае циклических сдвигов для периодичных строк, 2 метода решения.

4. (4) Поиск наибольшей общей подстроки k строк суффиксным массивом.

5. (4) Построение суффиксого дерева по суффиксному массиву и наоборот.

6. (3) Поиск подстроки в тексте суффиксным массивом за $O(m+\log n)$.

7. (4) Решение задач с помощью хешей: максимальный подпалиндром за $O(n)$, максимальный общий подпалиндром за $O(n\log n)$.

8. (3) Сравнение строк с помощью хешей за $O(\log n)$. Построение суффиксного массива за $O(n\log^2 n)$.

9. (3) Определение несжатого суффиксного дерева, наивное построение за $O(n^2)$. Методы хранение рёбер — массив, список, `set`, хеш, бор. Чит-метод хранения рёбер для несжатого суффиксного дерева (для каждой вершины храним или 1 ребро, или ссылку на массив).

10. (3) Реализация Укконена за $O(n^2)$.

11. (3) Сжатое суффиксное дерево. Построение за ($\texttt{time} = O(n^2), \texttt{memory} = O(n)$) (списками рёбер).

12. (5) Реализация Укконена за $O(n)$ (с доказательством времени работы).

13. (3) Решение задач с помощью суффиксного дерева: число подстрок, максимальная общая подстрока.

14. (3) Решение задач с помощью суффиксного дерева: максимальный подпалиндром, максимальный общий подпалиндром.

15. (3) КМП. Автоматы, автомат для КМП.

16. (3) Ахо-Корасик.

17. (3) Решение вероятностной задачи про мартышек методом Гаусса за $O(n^3)$.

18. (4) Построение суффиксного дерева по суффиксному автомату и наоборот.

19. (4) Разбиение строки на простые. Алгоритм Дюваля поиска минимального циклического сдвига.

20. (5) Алгоритм построения суффиксного автомата (с доказательством времени работы).

Потоки

1. (3) Мистер Поток, господин Разрез, товарищ Дополняющий путь, насыщенные и ненасыщенные рёбра, Остаточная сеть. Хранение Графа списком рёбер, e, $e \oplus 1$.
2. (3) Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона поиска потока и разреза.
3. (3) Декомпозиция потока (поиском в ширину и в глубину, на примере задачи о поиске K путей). Ершинные мистер Поток и господин Разрез, мистер Поток с есколькими токами и стоками (ливание ершин).
4. (4) Алгоритм Эдмондса-Карпа за $\mathrm{O}(VE^2)$ с доказательством.
5. (4) Алгоритм Масштабирования за $\mathrm{O}(E^2 \log Max)$ с доказательством. Реализация с `k-`. Проталкивание максимума по пути, $\mathrm{O}(E^2)$.
6. (4) Градиентный метод (двоичный поиск и поиск в грубину) за $\mathrm{O}(E^2 \log^2 Max)$. Использование Дейкстры.
7. (5) Алгоритм Диница за $\mathrm{O}(V^2E)$. Алгоритм Малхотры-Кумара-Махешвари (поиск потока за $\mathrm{O}(V^3)$).
8. (4) Скрещивание алгоритм Диница с масштабированием за $\mathrm{O}(VE \log Max)$.
9. (3) Поиск паросочетаний, минимальное контролирующее множество, максимальное независимое множество. Поиск паросочетаний через потоки. Алгоритм Куна.
10. (4) Алгоритм Хопкрофта-Карпа за $\mathrm{O}(E\sqrt{V})$.
11. (4) Венгерский алгоритм за $\mathrm{O}(n^4)$. Поиск паросочетания по нулевым ребрам.
12. (5) Венгерка за $\mathrm{O}(n^3)$.
13. (3) Поиск разреза минимального веса, контролирующего множества минимального веса, независимого множества максимального веса.
14. (3) Использования Дейкстры для поиска пути (потенциалы Джонсона).
15. (3) Использование «детского» Форд-Беллмана. Доказательство того, что поиск пути в худшем случае работает за $\mathrm{O}(VE)$.
16. (3) Использование потенциалов Джонсона для поиска пути между всеми парами вершин в произвольном графе за $\mathrm{O}(VE \log V)$.
17. (3) Теорема Холла.
18. (4) Алгоритм поиска максимального потока минимальной стоимости. Теорема об оптимальности полученного результата. Неполиномиальность алгоритма.
19. (4) Поток с нижними и верхними пропускными способностями на ребрах (через поток минимальной стоимости).
20. (4) Построение паросочетания: максимальный вес ребра $\to$ минимальный за $\mathrm{O}(VE)$ (без двоичного поиска).

Игры

1. (3) Симметричные игры на ациклических графах (динамической программирование за $\mathrm{O}(E)$).
2. (3) Симметричные игры на циклических графах (Ретроанализ за $\mathrm{O}(E)$).
3. (4) Функция Гранди. Определения через mex, $(mex = 0) \Leftrightarrow$ `LOSE`. Прямая сумма игр, магическая сила волшебного `xor`-а.
4. (3) Эквивалентность по Гранди всех ацикличных игр нимам.
5. (4) Задачи на функцию Гранди: Ним, но кучку можно делить на произвольное число слагаемых, Щтирлиц, Мюллер и очередь, Hacking Bush.
6. (5) Прямая сумма игр на циклических графах. Алгоритм Смита за $\mathrm{O}(VE)$.