Практика по алгоритмам #15
Конец

---

1. Новые задачи

1. Найти отрезок зацикленного массива минимальной длины с суммой хотя бы x за $\mathcal{O}(n)$. Возможны отрицательные числа.
2. Дерево Штейнера для множества вершин T – связный подграф, содержащий все вершины из T, такой, что суммарный вес всех ребер подграфа минимален. Наша задача – найти дерево Штейнера. Веса неотрицательны. Граф неориентирован.
   a) Для $|T| = |V|$ найти за $\mathcal{O}(E \log V)$
   b) Для $|T| = 2$ найти за $\mathcal{O}(E \log V)$
   c) Для $|T| = 3$ найти за $\mathcal{O}(E \log V)$
   d) Для $|T| = 4$ найти за $\mathcal{O}(V^3)$
   e) Для $|T| = k$ найти за $\mathcal{O}(V^3 + 3^k V)$
3. Задача про счётчик
   a) Изначально значение счётчика равно 0. Затем n раз происходит операция `+1`. Докажите, что амортизированное время работы одной операции не $\mathcal{O}(\log n)$, а $\mathcal{O}(1)$.
   b) В предыдущей задаче мы делали только операцию `+1`. Придумайте структуру, которая делает операции `+1`, `-1`, `isZero`, каждую за амортизированное $\mathcal{O}(\log^*(n))$.
   c) А теперь все те же самые операции за $\mathcal{O}(\log^*(n))$ в худшем случае.
4. В алгоритм A^* подсунуть функцию оценки f, на которой он будет работать дольше. Лучше даже экспоненциально долго.
5. Выразить количество `AVL` деревьев высоты h из n вершин через произведение многочленов.
6. Выбрать всем вершинам потенциалы так, чтобы веса всех ребёр совпали.
7. Найти второй по величине кратчайший путь в графе за $\mathcal{O}(E \log V)$.

2. Разбор задач

1. **Кратчайший отрезок с суммой хотя бы x**
   Насчитаем заранее частичные суммы $s_i = \sum[1..i]$. Храним текущий оптимальный ответ $best$, изначально $+\infty$. Перебираем правую границу отрезка r. Для каждого r нужно выбрать такое l: $l \le r \wedge s_r - s_l \ge x \wedge r - l \to \min$. Пусть на отрезке $[1..r)$ у нас есть последовательность минимумов: $s_{m_1} < s_{m_2} < s_{m_3} < \cdots < s_{m_k}$ $(1 \le m_1 < m_2 < m_3 \cdots < r)$. Пока $s_r - s_{m_1} \ge x$, то обновляем $best$ значением $r - m_1$ и удаляем m_1 из дека минимумом. В конце у нас есть желание добавить на вершину стека пару $\langle r, s_r \rangle$. Для этого, пока $s_{m_k} \ge s_r$, снимаем верхний элемент со стека, и в конце кладем на стек $\langle r, s_r \rangle$.
2. **Дерево Штейнера**
   a) $|T| = |V| \Rightarrow$ минимальное остовное дерево
   b) $|T| = 2 \Rightarrow$ кратчайший путь между вершинами
   c) $T = \{A, B, C\}$, значит дерево имеет вид центр M, а из него пути в A, B, C. Найдем Дейкстрой кратчайшие пути от A, B, C. Перебрем вершину M.
   d) $T = \{A, B, C, D\}$. Дерево имеет одну из трех конфигураций, перебираем, какую. Запустим в начале Флойда. Теперь нужно перебрать два центра – X, Y и проверить $d[X, Y] + d[A, X] + d[B, X] + d[C, Y] + d[D, Y]$ на оптимальность.

e) $|T| = k$. Запустим в начале Флойда. Решение задачи – динамика по подмножествам $f[v, A, type]$, v – корень дерева, которое мы строим, $A \subseteq [1..k]$ – вершины, которые нужно покрыть, $type$ – 0 или 1, f – суммрные вес уже выбранных рёбер. Из $[v, A, 0]$ делаем переход-расщепление $relax(f[v, A, 0], f[v, B, 1]+f[v, A\backslash B, 1])$. Из $[v, A, 1]$ для каждой вершины u делаем переход-путь $relax(f[v, A, 1], f[u, A, 0]+d[v, u])$. Время работы алгоритма $\mathcal{O}(V^3 + 3^k V + 2^k V^2)$. Если мы не добавили третий параметр $type$, получился бы граф с циклами, на котором динамику считать не получилось бы.

3. **Счетчик**

a) Когда мы i-й раз делаем `+1`, k_i единиц заменятся на 0, один ноль заменится на 1. Время работы t_i равно $k_i + 1$. За n запросов появится n единиц $\Rightarrow$ не более n раз единица заменится на ноль $\Rightarrow \sum k_i \le n \Rightarrow \sum t_i \le 2n$.

b) Рассмотрим систему счисления $x = \sum\limits_{i=0}^{\infty} d_i 2^i$, где $0 \le d_i \le 2$. Число x может иметь несколько представлений в такой "системе счисления". Сделаем `+1`: если нулевой разряд был равен 0 или 1, то время $\mathcal{O}(1)$, иначе мы двойку меняем на единицу и делаем `+1` в следующий разряд. Заметим, что все операции кроме последней порождают единицу, а последняя операция уменьшит число единиц не более чем на один. То же самое с `-1`. За n запросов удалится не более n единиц $\Rightarrow$ появится не более $n + \mathcal{O}(\log n)$ единиц $\Rightarrow$ время работы $\mathcal{O}(n)$. Операция `isZero`: нужно хранить количество ненулевых цифр d_i. Для этого предположим, что с числами не более $\log n$ операции `+1` и `-1` происходят за $\mathcal{O}(1)$.

c) Рассмотрим систему счисления: $x = \sum\limits_{i=0}^{\infty} d_i (2^{i+1} - 1)$, где $0 \le d_i \le 2$, но двойка может быть не более чем одна, а в более младших разрядах должны быть нули. Каждое число имеет единственное представление в таком виде, например $9 = 2 \cdot 1 + 7$, а $124 = 2 \cdot 15 + 31 + 63$, а $64 = 1 + 63$. Заметим, что операции `+1` и `-1` теперь можно сделать за $\mathcal{O}(1)$ без амортизации.. **P.S.** Заметим, что если такую систему счисления использовать в биномиальной куче, то мы получим кучу с `Add` за $\mathcal{O}(1)$, `Merge` и `ExtractMin` за $\mathcal{O}(\log n)$, все времена без амортизации. Если применить сверху `BootStrapping`, получается куча с `Add` и `Merge` за $\mathcal{O}(1)$, `ExtractMin` за $\mathcal{O}(\log n)$.

5. **AVL-деревья**

$f[h, n] = \sum\limits_{a=0}^{n-1} \Big[f[h\text{-}1, a] f[h\text{-}1, n\text{-}a\text{-}1] + 2 f[h\text{-}2, a] f[h\text{-}1, n\text{-}a\text{-}1] \Big]$.

А теперь увидим многочлены, пусть $f[h](x) = \sum f[h, n] x^n$, тогда $f[h](x) = x \cdot f[h-1](x) \cdot (f[h-1](x) + 2f[h-2](x))$. Многочлены мы умеем перемножать быстро Карацубой и Фурье.

6. **Потенциалы**

Ищем потенциалы p_v. Если все потенциалы увеличить на константу, веса рёбер не изменятся $\Rightarrow$ пусть $p_1 = 0$. Пусть мы угадали вес рёбер z, который мы получим в итоге. Возьмем любой остов графа и делая переходы от вершины 1 однозначно посчитаем все потенциалы по формуле: $z = w + p_a - p_b \Rightarrow p_b = (w + p_a) - z$. Теперь нужно для каждого ребра не из остова проверить, что вес этого ребра тоже равен z. Как угадать z? Потенциал $p_v = A_v \cdot z + B_v$, а рёбра не из остова задают линейные уравнения на z, которые нужно решить. Время работы $\mathcal{O}(n + m)$.