Первый курс, весенний семестр

Практика по алгоритмам #4

LCA, Euler Tour

Contents

1 Новые задачи

1. Есть бинарная**, возможно, не ассоциативная** функция f. Запросы: посчитать f на отрезке. $\langle \mathcal{O}(n^2), \mathcal{O}(1) \rangle$. Здесь и далее все задачи по умолчанию online, static.
2. Решите задачу gcd ± 1. $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$.
3. Дан массив из n чисел, но различных в нем не более $\sqrt{n}$, $|a_{i+1} - a_i| = 1$. Запросы: посчитать произведение на отрезке. $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$.
4. В алгоритме Фараха-Колтона-Бендера для хранения предподсчета на масках длины k мы тратили $\mathcal{O}(2^k k^2)$ памяти. Придумайте, как затратить $\mathcal{O}(2^k)$ памяти и времени, сохранив при этом сложность ответа на запрос.
5. Придумайте, как, используя только идею Sparse Table, отвечать на RMQ запросы за $\langle \mathcal{O}(n \log^* n), \mathcal{O}(\log^* n) \rangle$. Выпишите в явном виде рекуррентные соотношения для времени и памяти работы алгоритма. Две версии.
6. Пусть можно использовать $\mathcal{O}(n)$ предподсчета и отвечать на LCA запрос за $\mathcal{O}(d[a]+d[b])$. Напишите как можно более короткий код ответа на запрос LCA (код без учета предподсчета).
7. a) Для заданного фиксированного k посчитайте для всех вершин дерева `up[v, k]` за $\mathcal{O}(n)$.
   b) Используя умение из предыдущего пункта, решите LCA за $\langle \mathcal{O}(n), \mathcal{O}(\sqrt{n}) \rangle$.
8. Дано дерево, у каждой вершины есть вес.
   Запросы: минимум на пути из a в b. $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$.
9. Дан ориентированный граф, исходящая степень каждой вершины равна единице.
   Запросы: из вершины v сделать k шагов вперед.
   a) $\langle \mathcal{O}(n \log \max k), \mathcal{O}(\log k) \rangle$.
   b) $\langle \mathcal{O}(n \log n), \mathcal{O}(\log k) \rangle$.
10. Дано дерево и запросы: LCA; подвесить лист. $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$. Online.
11. Дано дерево, на вершинах которого могут быть пометки. Запросы: пометить вершину, снять пометку с вершины, число помеченных вершин в поддереве. $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$.
12. Даны два дерева из n вершин, вершины в обоих помечены числами от 1 до n без повторений. Запрос: по вершине из первого дерева и вершине из второго дерева сказать количество чисел от 1 до n, которые лежат в поддеревьях обеих вершин из запроса. $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$.
13. Рассмотрим Эйлеров обход дерева. Научиться с его помощью решать следующие задачи.
    a) Запросы: сделать корнем вершину v. $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$.
    b) (*) Много деревьев. Запросы: подвесть дерево с корнем v к вершине u другого дерева; отрезать поддерево с корнем в вершине v от ее дерева; в одном ли дереве лежат вершины u и v. $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$.
14. (*) Дан лес подвешенных деревьев. Запросы: LCA; подвесить дерево с корнем v к вершине u другого дерева. (a) Двоичными подъёмами за $\mathcal{O}(\log^2 n)$. (b) Эйлеровым обходом за $\mathcal{O}(\log n)$.
15. (*) Даны дуги на окружности. Найти max число попарно непересекающихся дуг. $\mathcal{O}(n \log n)$.
16. (*) Дано дерево, у каждой вершины есть вес. Запросы: изменить вес вершины; найти максимум на пути из u в v. $\langle \mathcal{O}(n \log n), \mathcal{O}(\log^2 n) \rangle$.
    Сначала полезно понять, что запрос на поддереве – это запрос на прямоугольнике.

2 Домашнее задание

2.1 Обязательная часть

1. (1) Запрос: найти расстояние (количество рёбер) между вершинами в дереве.
2. (2) Мы умеем искать LCA двоичными подъёмами за $\langle\mathcal{O}(n\log n),\mathcal{O}(\log n)\rangle$. Придумайте, как хранить разреженную таблицу двоичных подъёмов, чтобы решение работало за $\langle\mathcal{O}(n),\mathcal{O}(\log n)\rangle$. Скажите, как с помощью новой структуры считать значение произвольной ассоциативной функции на пути в дереве в online за $\langle\mathcal{O}(n),\mathcal{O}(\log n)\rangle$.
3. (2) Дан неорграф. Сделайте предподсчёт за линию на полилог, чтобы за $\mathcal{O}(\log n)$ в online отвечать на запросы вида "сколько вершин нельзя не пройти при путешествии из a_i в b_i?".
4. (2) Дан взвешенный неорграф. Найти максимальный по длине отрезок весов $0 \le l \le r \le M$ такой, что множество рёбер с весами из $[l,r]$ не содержит циклов. $\mathcal{O}((m+n)\log n)$.
5. (2) Дано дерево. Помечаются (удаляются) вершины. Нужно в online за $\langle n\log n,\log n\rangle$ отвечать на запрос самого нижнего не помеченного общего предка.
6. (2) Выбрать из массива длины n подпоследовательность длины L, в которой разность индексов соседних элементов не более k, а сумма элементов минимальна. Время $\mathcal{O}(nL)$.
7. (4) В дерево добавляются листья. Нужно быстро отвечать на запрос "размер поддерева".
   a) (2) Online с декартовым деревом.
   b) (2) Offline с деревом отрезков.

2.2 Дополнительная часть

1. (4) Дано дерево. Помечаются (удаляются) и разпомечаются (добавляются удалённые) вершины. Нужно в online за $\langle n\log n,\log n\rangle$ отвечать на запрос самого нижнего не помеченного общего предка.
2. (3) Дано подвешенное дерево из $n = 10^5$ вершин. В каждой вершине v стоит некоторая булева переменная x_v. У каждой вершины v есть булево значение $value_v$ (`true` or `false`). Значение переменной z в листе l, $f(z,l)$ определяется так:
   `for (v = l; v != root && x[v] != z; v = p[v]) ; return x[v] == z ?  value[v] :  -1;`
   Нужно обработать $m = 10^5$ запросов вида "присвоить вершине v значение $newValue$, и сказать у скольких листьев значение x_v равно `false`, у скольких `true`, у скольких `-1`".
3. (3) Дано дерево изначально пустых namespace-ов. Все переменные в задаче булевы. Чтобы, находясь в вершине дерева v узнать значение переменной x, нужно подниматься по дереву, пока не попадём в вершину, где x определена. Запросы:
   a) Добавить в namespace v запись `x := value`.
   b) Сказать, в скольки вершинах переменная x равна `true`.