SPb HSE, ПАДИИ, 1 курс, весна 2023/24
Практика по алгоритмам #17

RMQ/LCA

28 мая

Собрано 16 июня 2024 г. в 21:30

Содержание

RMQ/LCA

1. **Короткий код LCA**

   Пусть можно сделать $\mathcal{O}(n)$ предподсчета и отвечать на запрос $\mathrm{LCA}(a, b)$ за $\mathcal{O}(\mathrm{dist}(a, b))$.
   Напишите как можно более короткий код ответа на запрос LCA.
   Сам предподсчёт не пишите, его длину также не учитывайте.

2. **Что такое двоичные подъёмы?**

3. Посчитать длину пути в дереве за $\mathcal{O}(\log n)$.

4. Посчитать минимум весов вершин на пути в дереве за $\mathcal{O}(\log n)$.

5. **Динамические двоичные подъёмы**

   Дано дерево и запросы: LCA; подвесить лист. $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$.

6. **Что такое sparse table?** Придумываем динамику.

7. Насчитать gcd на отрезке.

8. **Улучшенный Sparse Table**

   Идея: давайте разобьём исходный массив на куски длины $\log n$, на каждом куске найдём минимум, на полученном массиве длины $\frac{n}{\log n}$ делаем sparse table. Как это довести до подсчёта минимума за какое-либо приличное время?

9. **LA в offline**

   a) Для заданного фиксированного k посчитайте для всех вершин дерева `up[v,k]` за $\mathcal{O}(n)$.
   b) Решите задачу LA в offline за $\mathcal{O}(n + m)$.

10. **Поддерево – это отрезок**

    Дано дерево. Запросы: пометить вершину, снять пометку с вершины, число помеченных вершин в поддереве. $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$.

Разбор задач практики

1. **Короткий код LCA**

```
int lca(int a, int b): // O(distance between a and b)
    while (depth[a] > depth[b]) a = parent[a];
    while (depth[b] > depth[a]) b = parent[b];
    while (a != b) a = parent[a], b = parent[b];
    return a;
```

2. **Что такое двоичные подъёмы?**

   См. слайды или конспект ПМИ.

3. Посчитать длину пути в дереве за $\mathcal{O}(\log n)$.

   `depth[a] + depth[b] - 2*depth[LCA(a,b)]`

4. Посчитать минимум весов вершин на пути в дереве за $\mathcal{O}(\log n)$.

   Двоичные подъёмы. Кроме `up`$[v, 2^k]$ вычисляем минимумы на пути от v до `up`$[v, 2^k]$. `minw[v, k] = min(minw[v, k-1], minw[up[v, k-1], k-1])`.

5. **Динамические двоичные подъёмы**

   Просто пересчитываем двоичные подъемы из новых листьев за $\mathcal{O}(\log n)$, ведь из их предков уже все посчитано.

6. **Что такое sparse table?**

   См. слайды или конспект ПМИ.

7. **Насчитать gcd на отрезке.**

   Sparse Table подходит и для gcd тоже.

8. **Улучшенный Sparse Table**

   См. конспект ПМИ про улучшения Sparse-Table.

9. **LA в offline**

   a) Обойдем дерево `dfs`, поддерживая в стеке `path` путь от корня до вершины. `up[v, k] = path[top - k]`.
   b) $\forall v$ сохраним $ids[v]$ – список запросов, в которых участвует вершина v. Воспользуемся `dfs` из предыдущего пункта. В вершине v для всех $i \in ids[v]$ можно записать `ans[i] = path[top - k`$_i$`]`, где k_i – насколько нужно подняться в i-м запросе.

10. **Поддерево – это отрезок**

    Возьмем Эйлеров обход дерева, в котором каждая вершина встречается один раз (например, перед входом в поддерево). Каждое поддерево соответствует отрезку обхода. Начало отрезка – позиция корня поддерева. Длина – размер поддерева.

    Отвечаем на запросы деревом отрезков.

Домашнее задание

1. **(2) Генеологическое дерево**

   Дано корневое дерево из n вершин-организмов, обозначающее «кто от кого произошёл» (все произошли от корня). Нужно сделать предподсчёт за $\mathcal{O}(n \log n)$, чтобы затем быстро отвечать на запросы вида «найти ближайшего общего предка для организмов $i_1, i_2, \dots i_k$».

   a) **(1)** за $\mathcal{O}(k \log n)$.
   b) **(1)** за $\mathcal{O}(k + \log n)$.

   При решении этой задачи предполагайте, что LCA вы умеете искать только за $\mathcal{O}(\log n)$. Если же кто-то так хочет пользоваться LCA за $\mathcal{O}(1)$, придётся писать код, показать знание на практике.

3.1. Дополнительная часть

1. **(2) Нельзя не пройти**

   Дан неорграф.
   Сделайте предподсчёт за линию на полилог, чтобы за $\mathcal{O}(\log n)$ в online отвечать на запрос $get(a_i, b_i)$: сколько *рёбер* нельзя не пройти при путешествии из вершины a_i в вершину b_i?