

LCA - Lowest Common Ancestor

FRANCISZEK PIETRUSIAK

Zadanie:

Dane jest drzewo o $n \leq 5 \cdot 10^5$; mamy szybko

odpowiedzieć na $q \leq 5 \cdot 10^5$ zapytań postaci:

↳ Podaj nr. wierzchołka, który leży na ścieżce z wierzchołka a do wierzchołka b .

Rozwiązanie:

Stworzymy tablicę $up[v][k]$, które dla każdego

wierzchołka v trzyma nr. wierzchołka, który

leży na ścieżce z korzenia do v oraz jest oddalony od v o dokładnie 2^k krawędzi. Jeśli nie ma takiego wierzchołka to $up[v][k] = 0$.

Def:

Przedkiem wierzchołka v jest wierzchołek, który leży na ścieżce z korzenia drzewa do v .

Zetliśmy, że mamy już wypełnioną tablicę up .

Teraz z dowolnego wierzchołka v jesteśmy w stanie w czasie $O(\log n)$ wyznaczyć jego przodków w dowolnej odległości od v .

Tak wyznaczyć tablicę up :

```
const int MxN = 1e6, LOG = 20; // ceil(log2(10^6)) = 20
vector<int> G[MxN];
int up[MxN][LOG], lvl[MxN];

// Przechodzimy po wszystkich wierzchołkach i ustalamy:
// - poziom na którym się znajdują (korzeń ma 1, kolejni synowie coraz większy)
// - ojca danego wierzchołka
void dfs(int v, int p)
{
    lvl[v] = lvl[p] + 1;
    up[v][0] = p;

    for (auto u : G[v])
        if (u != p)
            dfs(u, v);
}
```

w main'ie:

```
for (int i=1; i<=n; i++)
    if (up[i][0] == 0)
        dfs(i, i);

// dla każdego wierzchołka obliczamy kolejnych przodków w odległości  $2^k$ 
for (int k=1; k<LOG; k++)
    for (int x=1; x<=n; x++)
        up[x][k] = up[up[x][k-1]][k-1];
```

Jak znaleźć LCA?

```
int lca(int a, int b)
{
    // zakładamy, że a jest na większym poziomie (dalej od korzenia / niżej)
    if (lvl[a] < lvl[b])
        swap(a, b);

    // wyrównujemy poziomy a i b
    for (int i=LOG-1; i>=0; i--)
        if (lvl[a] - (1<<i) >= lvl[b])
            a = up[a][i];

    // jeśli a i b są tym samym wierzchołkiem to LCA(a, b) = a
    if (a == b)
        return a;

    // teraz kiedy a i b są na tym samym poziomie, to możemy skakać tak długo, aż ich ojcowie będą
    for (int i=LOG-1; i>=0; i--)
        if (up[a][i] != up[b][i])
        {
            a = up[a][i];
            b = up[b][i];
        }

    // zwracamy ojca a, który jest LCA(a, b)
    return up[a][0];
}
```

Zadanie do przerobienia:

↳ Komwojezer Bajtezer (IX 01, etap I)

↳ Randke (XIX 01, I etap)