Vähim ühine eellane – LCA

Marko Tsengov

14. jaanuar 2023. a.

1 k-s eellane

Ülesanne. Antud juurega puu n tipuga ning q päringut, kus igal päringul küsitakse mingi tipu a_i k-ndat eellast puus (eeldusel, et see eksisteerib).

Naiivne lahendus: võtame igal päringul tipu a_i ning leiame k korda selle vahetu eellase. Keerukus: $O(nq)$.

Ülearvutatud lahendus: enne päringuid arvutame *kuidagi* iga tipu iga eellase ning jätame selle meelde. Minimaalne keerukus: $O(n^2 + q)$, mälu $O(n^2)$.

„Kesktee“: enne päringuid arvutame iga tipu iga 2^h-nda eellase, mis eksisteerib (hajus tabel). Keerukus: $O(n \log n + q \log n) = O((n + q) \log n)$, mälukasutus $O(n \log n)$.

Kuidas leida $O(n \log n)$ ajas kõik sellised eelased? On selge, et rohkem, kui $\log_2 n$ sellist eellast pole vaja arvutada. Olgu 2^h eellane iga tipu jaoks, kus see eksisteerib, arvutatud, üritame leida 2^{h+1}-inda eellase. Selleks piisab leida tipu 2^h-is eellane ning selle tipu 2^h-is eellane, mis ongi algse tipu 2^{h+1}-is eellane.

Algoritm:

```
H = int(log2(n)) + 1
iga h jaoks 0..H:
    iga tipu t jaoks tippude loendis:
        kui eksisteerib tipu t 2**h-is eellane:
            p = t 2**h-is eellane
            kui eksisteerib tipu p 2**h-is eellane:
                t 2**(h+1)-is eellane = p 2**h-is eellane
```

Kui sellised eellased on välja arvutatud, tuleb nende abil saada kiiresti k-s eellane leida. Selleks saab näiteks minna alati üles maksimaalse kahe astme eellaste jagu, mis ei lähe kõrgemale, kui veel vaja minna on.

Algoritm:

```
no = algne tipp
k = kui palju minna üles
iga h jaoks H..0:
    kui 2**h <= k:
        kui tipu no 2**h-ndat eellast ei leidu:
            eellast ei leidu
        no = tipu no 2**h-is eellane
        k -= 2**h
```

Tasub märkida, et 2^h kujul arvud on kahendesituses parajasti üksikud bitid, seega saab algoritmis $2^h < k$ kontrolli ning k vähendamise asendada bitikontrollidega `(k & (1 << h)) != 0`.

2 Vähim ühine eellane

Ülesanne. Olgu antud juurega puu n tipuga ning q päringut, kus igal päringul küsitakse mingi kahe tipu a ja b vähimat (sügavaimat) ühist eellast puus (mis alati eksisteerib).

Naiivne lahendus: viime esmalt mõlemad tipud samale sügavusele puus, võttes sügavama tipu ning leides selle vahetu eellase, kuni tippude sügavus ühtib. Seejärel liigume mõlema tipuga ühe eellase võrra üles kuni tipud ühtivad, saadud tipp ongi otsitav. Keerukus: $O(nq)$.

k-nda eellasega lahendus: viime tipud sarnaselt naiivsele lahendusele samale sügavusele, kuid teades tippude sügavuse vahet saame seda teha k-nda eellase leidmisega $O(\log n)$ ajas. Kui nii leitud tipud ühtivad, on vähim ühine eellane leitud. Muidu saame vaadata järjest kahanevaid „hüppeid“ eellaste leidmiseks, teostades sellised „hüpped“, mis ei muuda tippe võrdseks. Kui kõik võimalikud hüpped on läbi vaadatud, on kummagi tipu otsene vanem otsitav vähim ühine eellane.

```
a, b = algsed tipud
kui a sügavus on suurem, kui b sügavus:
    vaheta a ja b

k = b ja a sügavuste vahe
b = tipu b k-s eellane

kui a ja b on võrdsed:
    tagasta a

iga h jaoks H..0:
    pa = a 2**h-is eellane
    pb = b 2**h-is eellane
    kui pa != pb:
        a = pa
        b = pb

tagasta a eellane
```