



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Puualgoritmit, osa 1/2

Algoritmit ongelmanratkaisussa,
kevät 2018





Johdanto

- *puu* on yhtenäinen, syklitön verkko, jossa on n solmua ja $n - 1$ kaarta
- jokaisen solmuparin välillä on yksikäsitteinen polku
- usein päätämme, että yksi solmuista on juuri, jonka alapuolelle muut solmut asettuvat
- tämän viikon avaintekniikka: tehokas nouseminen puussa



Oheismateriaalia

- Kisakoodarin käsikirja, luvut 18.1 ja 18.3
- <https://cses.fi/kkkk.pdf>



Tehokas eteneminen

- funktiolle f annetaan arvo $1 \dots n$ ja se palauttaa arvon $1 \dots n$
- haluamme selvittää, miten arvo x muuttuu, kun käytämme funktiota k kertaa
- esimerkiksi kun $k = 3$, haluamme laskea arvon $f(f(f(x)))$
- helppo ratkaisu vie aikaa $O(k)$, mutta haluamme tehokkaampaa



Tehokas eteneminen

- merkitään $f(x, k)$, miten x muuttuu, kun se menee k kertaa funktion läpi
- esimerkiksi $f(x, 1) = f(x)$ ja $f(x, 3) = f(f(f(x)))$
- idea: lasketaan etukäteen kaikki arvot muotoa $f(x, 2^i)$ sopivaan rajaan asti
- hyödyllinen kaava: $f(x, 2^i) = f(f(x, 2^{i-1}), 2^{i-1})$



Tehokas eteneminen

- kun haluamme sitten laskea arvon $f(x, k)$, jaamme k :n summaksi 2:n potensseja
- esimerkiksi jos $k = 11$, niin $k = 8 + 2 + 1$
- voimme siis laskea $f(x, 11) = f(f(f(x, 8), 2), 1)$
- summa muodostuu $O(\log k)$ osasta, joten laskeminen vie aikaa vain $O(\log k)$



Tehokas nouseminen

- tehokas eteneminen juurellisessa puussa: voimme nousta ylöspäin k askelta tehokkaasti
- nyt $f(x, k)$ ilmaisee, mihin solmuun tulemme, kun nousemme k askelta solmusta x
- pystymme vastaamaan mihin tahansa tällaiseen kyselyyn ajassa $O(\log n)$



Alin yhteinen esivanhempi

- kahden solmun *alin yhteinen esivanhempi* on alin puussa oleva solmu, joka on kummankin solmun esivanhempi
- merkitään, että $\text{lca}(a, b)$ on solmujen a ja b alin yhteinen esivanhempi
- voimme laskea minkä tahansa arvon $\text{lca}(a, b)$ ajassa $O(\log n)$ tehokkaalla nousemisella



Alin yhteinen esivanhemp

- vaihe 1: siirrämme alemmaa solmua niin, että molemmat solmut tulevat samalle tasolle
- vaihe 2: siirrämme rinnakkain solmuja mahdollisimman ylös kuitenkin päätyttä samaan solmuun
- tämän jälkeen solmuilla on yhteinen vanhemp, joka on alin yhteinen esivanhemp
- molemmat vaiheet vievät aikaa $O(\log n)$
- poikkeustapaus: toinen tai kumpikin solmu on valmiiksi alin yhteinen esivanhemp



Solmujen etäisyydet

- voimme myös laskea tehokkaasti, mikä on kahden solmun etäisyys puussa
- päätämme ensin, että jokin solmu on puun juuri
- tämän jälkeen voimme laskea solmujen a ja b etäisyyden kaavalla

$$\text{depth}[a] + \text{depth}[b] - 2 \cdot \text{depth}[\text{lca}(a, b)]$$

- voimme esikäsittelyn jälkeen laskea minkä tahansa etäisyyden ajassa $O(\log n)$