Lastnosti kopice

1 Splošno

Ker je kopica celovito drevo, velja

$$2^h \leq n \leq 2^{h+1} - 1$$

$$h = \lceil \lg(n+1) \rceil - 1 = \lfloor \lg n \rfloor$$

Gradnja kopice. Elemente je potrebno vsaj prebrati, zato je spodnja meja zahtevnost $\Omega(n)$.

2 Gradnja kopice z zaporednim vstavljanjem

Algoritem gradnje kopice z vstavljanjem elemente zaporedoma vstavlja v kopico. Pri tem se element doda na konec kopice, s čimer se kopica lahko pokvari, zato se nato izvede še dvigovanje elementa.

2.1 Najslabši primer

V najslabšem primeru se vsak element vedno dvigne na vrh. Takšen primer je npr. pri gradnji max-kopice, če so elementi podani v naraščajočem vrstnem redu. Vsak novi element je večji od vseh ostalih, zato se mora dvigniti do korena kopice. Vsako dvigovanje do vrha zahteva kvečjemu toliko zamenjav, kot je višina kopice. Višina končne kopice je $h = \lfloor \lg n \rfloor$, dvigniti pa je potrebno vseh n elementov, torej je skupna zahtevnost gradnje $O(n \lg n)$.

Natančno izpeljavo zahtevnosti pa naredimo takole. Elementi prihajajo po vrsti, pri čemer se i-ti dvigne za $\lfloor \lg i \rfloor$ nivojev. Če pogledamo na celotno zaporedje, se prvi element ne dvigne nič, nato dva za en nivo, nato štrije za dva nivoja, osem za tri nivoje, itd. V splošnem se po 2^i elementov dvigne za i nivojev. Predpostavimo, da je elementov enako $n = 2^{h+1} - 1$, s tem dobimo popolno kopico. Vse to seštejemo in dobimo

$$\sum_{i=0}^{h} i2^i = (h-1)2^{h+1} + 2 = (\lfloor \lg n \rfloor - 1)(n+1) + 2 = n\lfloor \lg n \rfloor - n + \lfloor \lg n \rfloor + 1 \sim n \lg n.$$

Zgornja meja časovne zahtevnosti je torej $O(n \lg n)$.

2.2 Povprečna zahtevnost

Oglejmo si še povprečno zahtevnost gradnje. Vprašajmo se, kolikšno je povprečnem število dvigov, ki jih naredi vozlišče na i-tem nivoju. Predpostavimo, da so vsi elementi različni in v vhodnem zaporedju enakomerno porazdeljeni. Tako je pri primerjanju dveh elementov enako verjetno, da je prvi večji kot da je drugi večji. Torej je verjetnost $\frac{1}{2}$, da dvig ni potreben, in $\frac{1}{2}$, da je potrebnen vsaj en dvig. Z nivoja i tako pridemo na nivo $i-1$. Označimo z $\bar{d}_i$ povprečno število dvigov elementa na nivoju i in že lahko zapišemo rekurenčno enačbo

$$\bar{d}_i = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot (1 + \bar{d}_{i-1}) = \frac{1}{2}(1 + \bar{d}_{i-1}),$$

pri čemer za prvi element oz. nivo $i = 0$ velja $\bar{d}_i = 0$.

Zapišimo nekaj členov $\bar{d}_i : 0, 1/2, 3/4, 7/8, 15/16, \ldots$, hitro opazimo, da je splošni člen enak

$$\bar{d}_i = \frac{2^i - 1}{2^i} = 1 - 2^{-i}.$$

Še dokažimo to z uporabo indukcije. Pri $i = 0$, velja $\bar{d}_i = 0$ in tudi $1 - 2^0 = 0$. Naredimo še induktivni korak $i - 1 \to i$. Torej

$$\frac{1}{2}(1 + \bar{d}_{i-1}) = \frac{1}{2}(1 + 1 - 2^{-(i-1)}) = 1 - 2^{-i}.$$

Podobno kot v najslabšem primeru velja, da 2^i elementov nivoja i v povprečju dvignemo za $\bar{d}_i$ nivojev. Predpostavimo, da je kopica polna oz. $n = 2^{h+1} - 1$. Torej je povprečno število vseh dvigov vseh elementov enako

$$\bar{d} = \sum_{i=0}^{h} 2^i \bar{d}_i = \sum_{i=0}^{h} 2^i (1 - 2^{-i}) = \sum_{i=0}^{h} (2^i - 1) = (2^{h+1} - 1) - (h+1) = (n+1) - 1 - \lg(n+1) = n - \lg(n+1) \sim n.$$

Povprečna zahtevnost je torej za faktor $\lg n$ boljša od zahtevnosti v najslabšem primeru. Rečemo lahko, da je povprečna zahtevnost $\Theta(n)$.

3 Gradnja kopice iz celotnega zaporedja

Ta način gradnje potrebuje vse elemente v naprej. Elementi so podani kot zaporedje v polju. Z zaporednim ugrezanjem notranjih vozlišč drevesa se polje postopoma spreminja v kopico.

3.1 Najslabši primer

V najslabšem primeru je potrebno vsak element ugrezniti do dna. Takšen primer je npr. pri max-kopici, kadar so elementi podani v naračajočem vrstnem redu. Vsako spuščanje zahteva torej do $\lg n$ ugrezov, skupna zahtevnost je potemtakem $O(n \lg n)$. Če pa naredimo malce bolj natančno analizo, lahko podamo precej bolj tesno oceno.

Listov seveda ni potrebno ugrezati. Nek element lahko ugreznemo le tolikokrat, kot je njegova višina. V splošnem za elemente na višini i potrebujemo i ugrezov.

V popolnem drevesu je višino vozlišča dobimo, če od višine drevesa odštejemo globino vozlišča. Elementov na globini i je 2^i. Torej lahko zapišemo

$$\begin{aligned}\sum_{i=0}^{h}(h-i)2^i &= h\sum_{i=0}^{h}2^i - \sum_{i=0}^{h} i2^i = h(2^{i+1}-1) - ((h-1)2^{h+1}+2) = h2^{h+1} - h - h2^{h+1} + 2^{h+1} - 2 = 2^{h+1} - h - 2 \\ &= (n+1) - (\lg(n+1)-1) - 2 = n - \lg(n+1) \sim n.\end{aligned}$$

Gradnja kopice na ta način ima torej zahtevnost $\Theta(n)$, kar je optimalno (najboljše možno). V primerjavi z zgornjim načinom pa je ta način gradnje v najslabšem primeru asimptotično gledano od zgornjega boljši za faktor $\lg n$.

Opomba. Rezultat je pravzaprav izredno zanimiv, ker sta na prvi pogled oba načina igradnje zelo podobna (nekje gre za dvigovanje, drugje za ugrezanje). Če pa podrobneje razmislimo, pa ugotovimo naslednje. V prvem načinu z naraščanjem števila elementov na nivoju narašča tudi zahtevnost dvigovanja elementov (torej oboje hkrati). V drugem načinu pa z naraščanjem števila elementov na nivoju, pada zahtevnost ugrezanja elementov. To pa očitno prispeva cel faktor k zahtevnosti gradnje.

3.2 Povprečna zahtevnost

Zaradi spodnje meje problema $\Omega(n)$ in najslabše zahtevnosti $O(n)$, seveda, ni za pričakovati nič drugega kot $\Theta(n)$. Vseeno pa je izpeljava zanimiva. Postopamo podobno kot zgoraj. Povprečno število ugrezov, ki jih naredi vozlišče na i-ti globini oz. na $(h-i)$-ti višini (imamo popolno drevo) je

$$\bar{u}_i = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot(1+\bar{u}_{i+1}) = \frac{1}{2}(1+\bar{u}_{i+1}).$$

Za zadnji nivo $i=h$ velja $\bar{u}_h = 0$. Za $\bar{u}_i$ od h proti 0 dobimo enako zaporedje kot zgoraj za $\bar{d}_i$ od 0 proti h, torej $\bar{u}_i = \bar{d}_{h-i}$. Nadalje

$$\begin{aligned}\bar{u} &= \sum_{i=0}^{h}2^i\bar{u}_i = \sum_{i=0}^{h}2^i\bar{d}_{h-i} = \sum_{i=0}^{h}2^i(1-2^{-(h-i)}) = \sum_{i=0}^{h}2^i - 2^{-h}\sum_{i=0}^{h}4^i = 2^{h+1} - 1 - 2^{-h}\frac{4^{h+1}-1}{3} \\ &= \frac{3\cdot 2^{h+1} - 3 - 2\cdot 2^{h+1} + 2^{-h}}{3} = \frac{1}{3}(2^{h+1} + 2^{-h} - 3) \\ &= \frac{1}{3}(n+1+\frac{2}{n+1} - 3) = \frac{n}{3} - \frac{2n}{n+1} \sim \frac{n}{3}.\end{aligned}$$