

A. subalg. recursiv $\Theta(\log_2 n)$

Subalgoritm ceva (n) este

dacă $n \leq 1$ atunci

ceva $\leftarrow 0$

altfel

ceva \leftarrow ceva ($\lfloor n/2 \rfloor$)

sf dacă

sf Subalgoritm

Justificare complexitate:

$$T(n) = T(n/2) + 1$$

$$\text{Notăm } n = 2^k \Rightarrow k = \log_2 n$$

$$T(2^k) = T(2^{k-1}) + 1$$

$$T(2^{k-1}) = T(2^{k-2}) + 1$$

\vdots

$$T(1) = 1$$

$$\textcircled{+} T(2^k) = \underbrace{1 + 1 + 1 + \dots + 1}_{\text{de } k \text{ ori}} \Rightarrow T(2^k) = k \Rightarrow$$

$$\Rightarrow T(n) = \log_2 n \Rightarrow \text{Complexitatea este } \Theta(\log_2 n)$$

(având în vedere că:

caz fav = caz defav = caz medie)

B. Înălțimea nodului 22, lungimea drumului cel mai lung de la nodul 22 la o frunză, este 3 (drumul: 22, 20, 50, 21).

↑ înălțimea reprezintă cel mai lung drum de la nod la o frunză, considerând că ne aflăm în subgraful în care nodul e rădăcină.

C1. Ștergerea unui element e dintre-un vector ordonat se poate face în $O(n)$, căutarea putându-se totuși realiza în $O(\log_2 n)$ prin algoritmul de căutare binară. Totuși, ștergerea efectivă a elementului ar necesita mutarea acelor elemente plasate după el, dacă se dorește păstrarea ordinii. În caz contrar, ștergerea se poate realiza în $O(\log_2 n)$ înlocuind elementul șters cu ultimul din vector (dacă nu e lex același elementul șters).

C2. Ambele referințe se modifică la adăugarea într-o coadă $viDA$, deoarece, după efectuarea operației, coada va conține un singur element: acesta este și primul și ultimul.

1. Pentru a realiza funcția de ștergere, vom elimina din ansamblu valoarea rădăcinii și vom pune în locul ei valoarea ultimului element din ansamblu. Cel mai probabil, acum nu se va mai respecta relația dată \Rightarrow este necesar să colocăm elementul mutat, până când sunt satisfăcute toate condițiile.

Ansamblu:

$e: TElem[]$

$n: \text{întreg}$

$r: \text{Relatie}$

Funcția ștergere (a) este

$\{pre: a \in \text{Ansamblu}$

$post: \text{din } a \text{ a fost eliminat elementul cu cea mai mare prioritate } \}$

$\{ \text{se înlocuiește primul cu ultimul element} \}$

$a.e[1] \leftarrow a.e[a.n]$

$a.n \leftarrow a.n - 1$

$\{ \text{se va coloca elem. din rădăcină până când îndeplinește} \}$

$\{ \text{cond. de ansamblu} \}$

$poz \leftarrow 1$

$poz - d \leftarrow poz * 4$

Cât timp $\text{poz} - d \in a.m \wedge a.n(a.e \in \text{poz} - d, a.e \in \text{poz})$ execută

2 vom alege cel mai prioritar descendent ?

dacă $\text{poz}^{*4+1} \in a.m \wedge a.n(a.e \in \text{poz}^{*4+1}, a.e \in \text{poz} - d)$
atunci

$$\text{poz} - d \leftarrow \text{poz}^{*4+1}$$

SR Dacă

Dacă $\text{poz}^{*4+2} \in a.m \wedge a.n(a.e \in \text{poz}^{*4+2}, a.e \in \text{poz} - d)$
atunci

$$\text{poz} - d \leftarrow \text{poz}^{*4+2}$$

SR Dacă

Dacă $\text{poz}^{*4+3} \in a.m \wedge a.n(a.e \in \text{poz}^{*4+3}, a.e \in \text{poz} - d)$
atunci

$$\text{poz} - d \leftarrow \text{poz}^{*4+3}$$

SR Dacă

3 după ce am ales descendentul cel mai prioritar ?
3 va trebui să aflăm elem de pe poz ?

$$aux \leftarrow a.e \in \text{poz}$$

$$a.e \in \text{poz} \leftarrow a.e \in \text{poz} - d$$

$$a.e \in \text{poz} - d \leftarrow aux$$

} acum vom face aceeași verificare și pt. următorii }

↓ descendenți }

$$pow \leftarrow pow - d$$

$$pow_d \leftarrow pow * 4$$

↓ se înmulțește cu 4, deoarece este ansamblu cuaternion

Se Căutăm

Se Funcție

Complexitate:

- de timp: $O(4^n \cdot m)$
4ⁿ ← fiind ansamblu cuaternion
- de spațiu (suplementar): $\Theta(1)$