

Heap-uri. Exerciții rezolvate

1. Care este numărul maxim de chei, care pot fi stocate într-un heap-max (min) de înălțime h ?

Soluție: Numărul maxim de chei se obține atunci când toate nivelurile sunt complete. Acest lucru se poate vedea în figura 1.

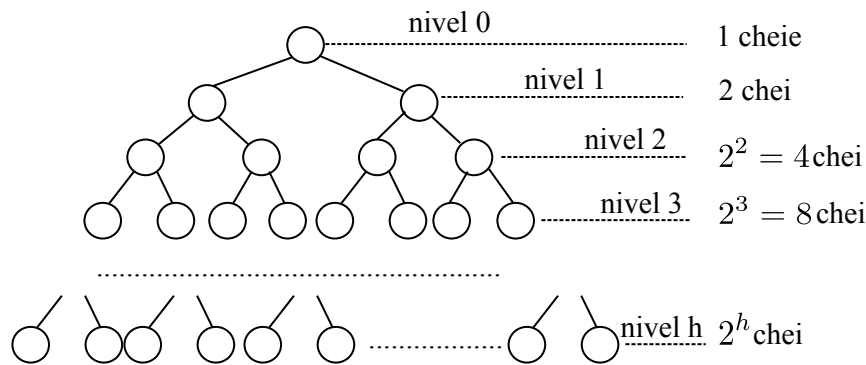


Figure 1: Heap max cu înălțimea h și număr maxim de chei.

Pe nivelul 0 avem o singură cheie, pe nivelul 1 avem 2 chei, pe nivelul 2 sunt $2^2 = 4$ chei. Inductiv se poate arăta că pe nivelul k sunt în acest caz 2^k chei.

Astfel numărul maxim de chei în heap este:

$$n = 1 + 2 + 2^2 + \dots + 2^h = 2^{h+1} - 1$$

2. Care este numărul minim de chei, care pot fi memorate într-un heap-max de înălțime h ?

Soluție: Să nu uităm faptul că, un heap este un arbore complet. Ceea ce înseamnă că, înainte de a trece la un nivel nou, nivelul anterior trebuie să fie complet. Astfel numărul minim de chei se obține atunci când, toate nivelurile până la penultimul ($h-1$) sunt pline, iar pe ultimul nivel avem o singură cheie. Acest lucru este ilustrat în fig. 2.

În acest caz numărul de chei va fi:

$$n = 1 + 2 + 2^2 + \dots + 2^{h-1} + 1 = 2^{(h-1)+1} - 1 + 1 = 2^h$$

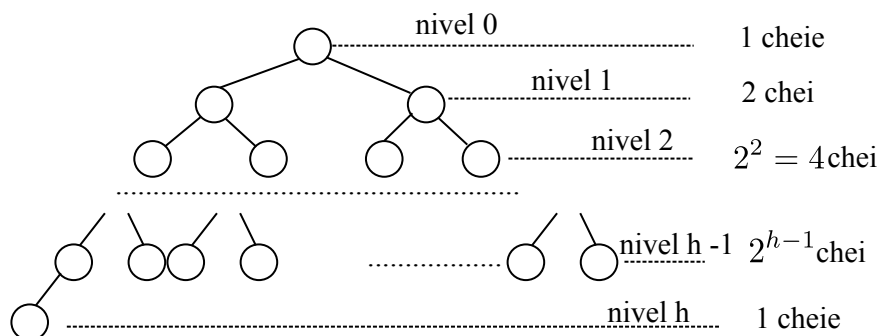


Figure 2: Heap max cu înălțimea h și număr minim de chei.

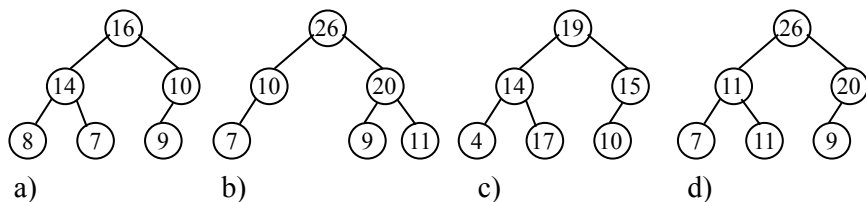


Figure 3:

3. Care dintre următorii arbori binari (fig. 3) au structura de heap-max?

Soluție: a) Este heap max, deoarece este arbore complet și pentru fiecare nod x , copiii săi au cheile mai mici decât cheia nodului x .

b) Nu este heap max, deoarece nu are structura de arbore complet. Pe nivelul 2 lipsește un fiu al nodului 10.

c) Nu este heap max. Are structura de arbore complet, dar fiul drept al lui 14 are cheia 17, care este mai mare decât 14.

d) Este heap max, deoarece este arbore complet și pentru fiecare nod x , copiii săi au cheile mai mici (sau egale) decât cheia nodului x . Putem avea noduri cu chei egale!

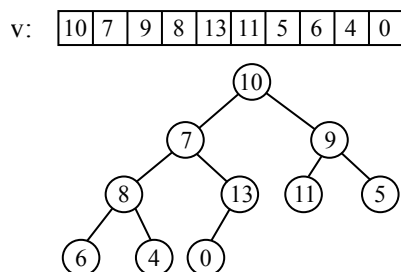


Figure 4:

4. Se consideră vectorul $v = \{10, 7, 9, 8, 13, 11, 5, 6, 4, 0\}$. Este heap-max? Dacă nu, să se transforme în heap max. Din heap-ul obținut, să se extragă maximum.

Soluție: Arborele complet reprezentat de vectorul v este cel din figura 4.

Se observă faptul că: nodul cu cheia 7 are ambii copii cu chei mai mari (8 și 13) și nodul cu cheia 9 are copilul stâng cu cheie mai mare (11). Deci nu este heap max. Aplicăm algoritmul din curs CONSTRUCT_HEAP, ilustrat în fig. 5. Se pornește de la primul nod, care are copii, și anume nodul 13. Copiii lui 13 sunt ambii mai mici, deci se trece la nodul 8. Și în cazul acestuia, copiii sunt mai mici. Se ajunge la nodul 9. În cazul acestui nod, fiul stâng (11) este mai mare. Se interschimbă 11 cu 9 (se aplică funcția MAX_HEAP). Apoi se trece la nodul 7. În acest caz ambii fii sunt mai mari. Se alege fiul cu cheia cea mai mare (13) și se interschimbă cu 7 (se aplică HEAP_MAX pentru a coborâ 7 pe poziția potrivită). În acest moment 7 are ca fiu pe 0, care este mai mic, și se oprește algoritmul HEAP_MAX. Se ajunge la nodul din vârf, cu cheia 10. Se observă că are fiul stâng, 13, mai mare. Se aplică funcția MAX_HEAP și de coboară nodul 10 pe poziția potrivită.

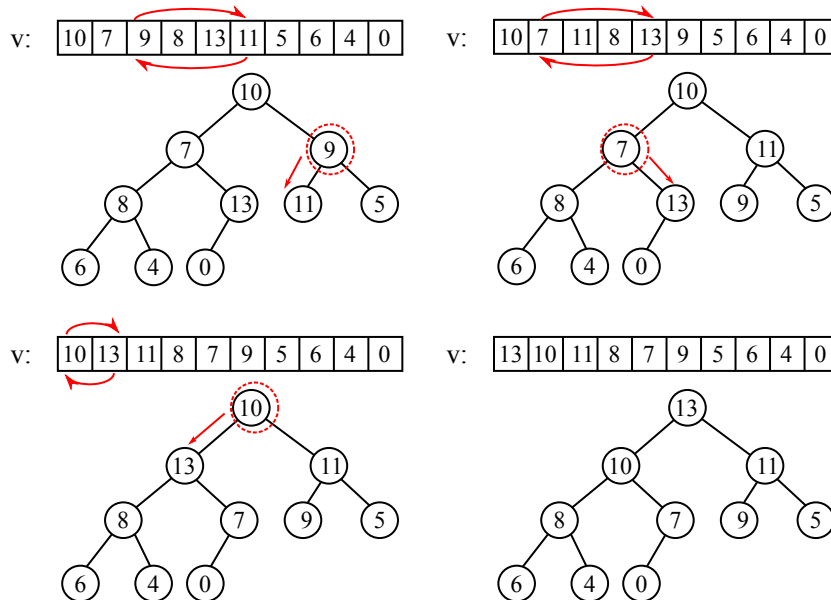


Figure 5: Construcția heap-ului max conform algoritmului CONSTRUCT_HEAP

Extragem maximul din heap-ul obținut. Pentru aceasta se plasează ultima cheie din heap, 0, pe prima poziție, iar apoi se reface heap-ul, prin coborârea lui 0 pe poziția potrivită (HEAP_MAX). Acest proces este ilustrat în figura 6.

5. Să se insereze pe rând într-o coadă de priorități inițial vidă cheile: 3, 10, 7, 15, 9, 11, 17, 8. Soluția este încărcată în fișierul de exemplificare al inserției pe pagina de elearning.

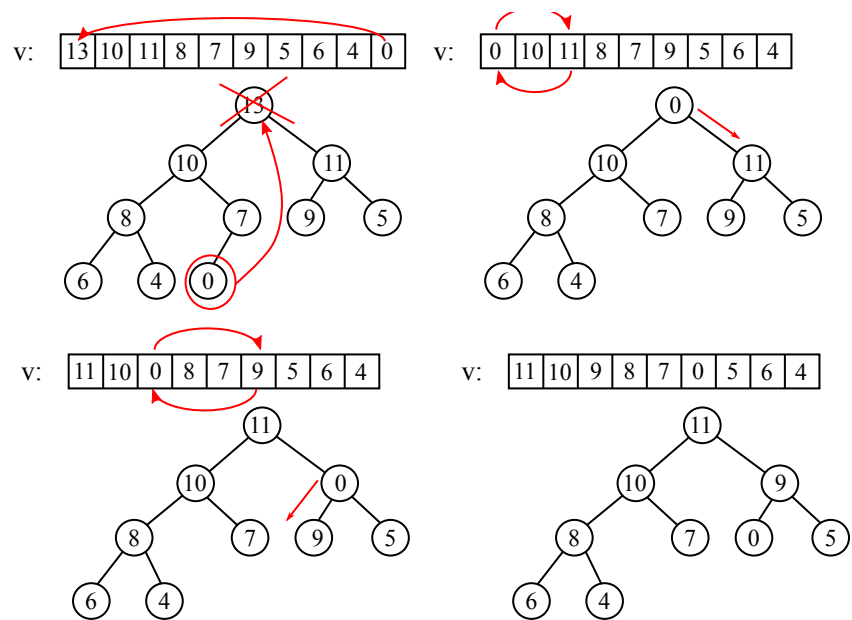


Figure 6: Extragerea maximului dintr-un heap max.