

1. Odredite vremensku složenost datog fragmenta programskog kôda. Broj koraka prikazati u obliku polinomijalnog izraza i u O notaciji.

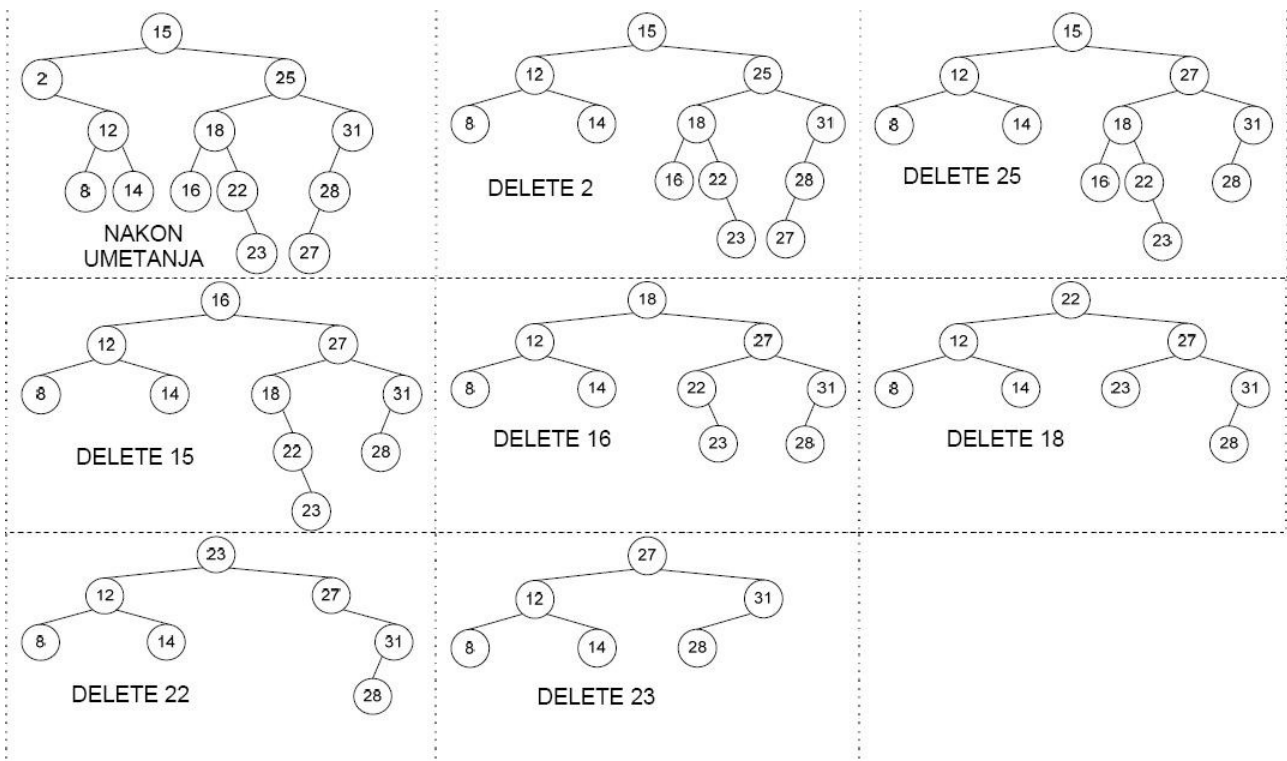
```
for (i=n;i>0;i>>=1)
    if (i%2==0) a[i]=1;
```

Rešenje: Kako se iz ciklusa izlazi kada polazno  $i=n$  postane jednako 0, jasno je da tražimo  $T(n)$ .

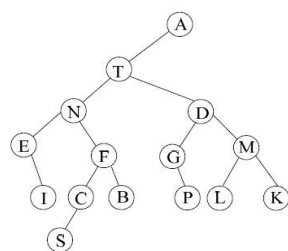
$$T(n) = 1_{(i=n)} + 1_{(i>0)} * ([\log n] + 2) + 1_{(i\%2=0)} * [\log n]$$

2. U binarno stablo pretrage najpre se redom umeću ključevi 15, 25, 18, 31, 16, 2, 12, 14, 8, 22, 23, 28, 27. Zatim se redom brišu ključevi 2, 25, 15, 16, 18, 22, 23. Prikazati izgled stabla nakon svih umetanja i nakon svakog brisanja.

Jedno od rešenja:



3. Ako za jedno binarno stablo **preorder** obilazak daje poredak ATNEIFCSBDGPMLK, a **inorder** obilazak daje poredak EINSFBTGPDLMK, skicirati izgled ovog stabla, ako se zna da svaki čvor stabla sadrži jedno veliko slovo abecede.



4. Prikazati sadržaj heš tabele nakon smeštanja ključeva redom 70, 26, 37, 59, 99, 24, 2, 12, 42, 31. Pretpostavite da je tabela veličine  $n=11$ , da je primarna hash funkcija  $h_1(x) = x \% 11$ , kao i da se za razrešavanje kolizija koristi otvoreno adresiranje sa dvostrukim heširanjem sekundarnom heš funkcijom  $g(x) = 1 + x \% 10$ .

0	1	2	3	4	5	6	7	8	9	10
26	37	24	59	70	2		12	31	42	99

5. Precizno opisati algoritam heapsort i oceniti njegovu složenost u najgorem slučaju. Prikazati rad algoritma po koracima pri sortiranju niza: 57, 42, 69, 11, 35, 28, 7, 19 u neopadajućem poretku.

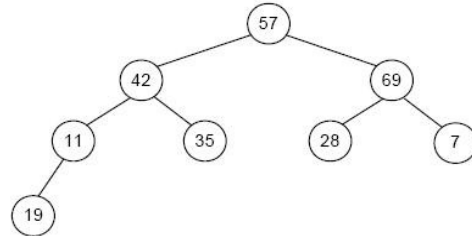
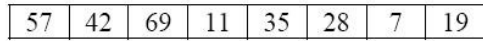
Opis algoritma:

- 1) Neuređen niz se najpre preuredi u hip. Sve izmene se vrše nad nizom koji se uređuje, pa se ne koristi dodatni prostor. Preuređivanje u hip se vrši zamenom mesta elemenata niza, počevši od prvog elementa niza (odnosno korena stabla) tako da budu ispoštovani uslovi odnosa vrednosti roditelj-potomak koji važe u hipu.
- 2) Sve dok se ne obradi svih  $n$  elemenata ulaznog niza raditi sledeće

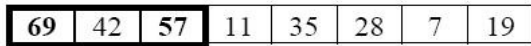
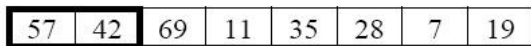
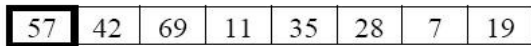
- vrednost u korenu menja mesto sa poslednjim elementom nesortiranog dela niza, odnosno onim delom niza koji je još uvek u hipu
- nova vrednost u korenu se propagira niz stablo sve dok ne uspostavi odnos roditelj-potomak koji važi u hipu
- Vremenska složenost je generalno  $O(n \log n)$ , a najbolji, najgori i prosečan slučaj se razlikuju za multiplikativnu konstantu.

Formiranje hipa:

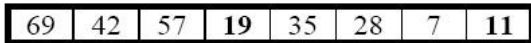
Na osnovu ranijeg primera, stablo koje se dobija od zadatog niza je prikazano na sledećoj slici. Ono, međutim ne zadovoljava uslove hipa (desni potomak čvora sa ključem 57 sadrži ključ 69 što nije dozvoljeno; slično važi i za ključeve 11 i 19), što se ne može jednostavno videti u datom nizu.



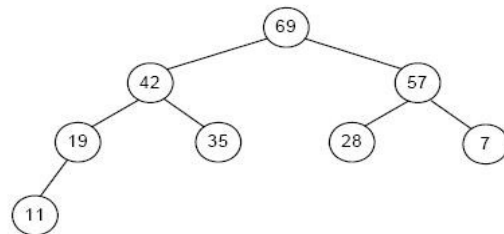
Preuređivanje niza počinje od prvog elementa niza (korena) i redom prolazi kroz sve elemente niza i po potrebi zamenjuje mesta datom elementu i elementu u njegovom roditeljskom čvoru, kao što je to prikazano na sledećoj sekvenci:



...



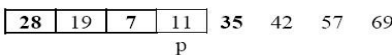
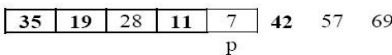
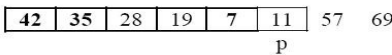
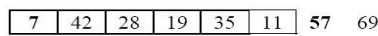
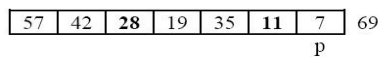
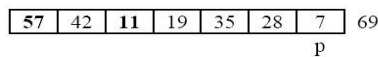
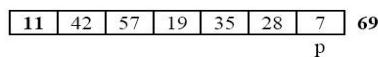
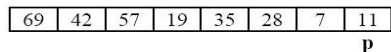
Rezultujuće stablo:



Sortiranje niza:

Ključ koji se nalazi u korenu (A) razmenjuje mesto sa poslednjim ključem u nesortiranom nizu (P). Čvor gde se smešta ključ (A) predstavlja sortirani deo niza i izuzima se iz daljeg razmatranja (grafički se to predstavlja njegovim razvezivanjem od ostatka stabla). Za ključ (P), koji se smešta u koren stabla, mora da se proveriti da li zadovoljava uslove hipa. Ako ne zadovoljava, taj ključ se propagira ka listovima stabla, a propagacija se završava kada su uslovi hipa zadovoljeni.

Sledeće slike ilustruju postupak sortiranja:



Ključ u korenu (69) menja mesto sa ključem na kraju nesortiranog dela niza (indeksira se promenljivom p).

Nakon smeštanja ključa na kraj niza, indeks p se smanjuje za 1. Ključ 69 se sada nalazi u sortiranom delu niza.

Ključ koji je smešten u koren stabla je manji od ključeva u sinovima korena. Tada on menja mesto sa većim ključem (u ovom slučaju ključem 57).

Slična situacija je i sa ključem 28.

Nakon ove izmene, zadovoljeni su uslovi hipa u okviru nesortiranog dela niza.

Ključ u korenu zamenjuje mesto sa ključem na kraju nesortiranog dela niza.

Nakon preuređivanja niza da bi se zadovoljili uslovi hipa, dobija se ovakva situacija

6. Napisati program (na programskom jeziku C) vremenske složenosti  $O(n)$  koji učitava sa standardnog ulaza niz realnih brojeva  $a$ , dimenzije  $n$  i pronalazi i ispisuje na standardni izlaz član niza čiji broj pojava u nizu je veći od  $n/4$ . Obrazložiti vremensku složenost konstruisanog rešenja.