

Rješenja 1. Međuispita iz ak. god. 2011./2012.

(Zadaci 1-4)

1. zadatak

Podsjetimo se prvo nekoliko teorijskih osnova *skip* (preskočnih) lista koje su potrebne za rješavanje ovog zadatka. Svaki čvor je definiran svojom vrijednosti (ključem) i razinom. Razina čvora određuje koliko će pokazivača sadržavati dani čvor. Ti pokazivači pokazuju na slijedeće čvorove u njihovoj pripadnoj razini. Uzmimo za primjer neki pokazivač u razini 3 – prateći njega možemo preskočiti sve čvorove niže od njegove razine.

U teoriji imamo definirano nekoliko iznosa i pravila:

n – predviđeni kapacitet liste

p – vjerojatnost prelaska čvora u višu razinu

$P(k) = p^{k-1}(1-p), k \in \mathbb{N}$ – vjerojatnost da čvor postigne k -ti stupanj

$E(n_k) = nP(k) = np^{k-1}(1-p)$ – očekivani broj čvorova u k -tom stupnju

$h = \left\lceil 1 + \log_{\frac{1}{p}} n \right\rceil$ – najviši stupanj čvora u listi

Prijeđimo na zadatak:

(7) Skicirajte izgled (strukturu) *skip* liste čiji su parametri $n=8$ i $p=0.5$ nakon što u polazno praznu listu upišemo redom brojeve: 16, 8, 3, 13, 1, 15, 2 i 4, uz pretpostavku da se primjenjuje učinkovitiji način određivanja stupnja novog čvora te da je slijed nasumičnih brojeva izračunanih tijekom postupka: 0.81, 0.99, 0.69, 0.91, 0.68, 0.42, 0.09, 0.18 i 0.13.

Uz dane brojeve, predstavljene su nam i slučajne vrijednosti. U rasponu su $(0,1)$, te ih vrlo praktično možemo iskoristiti za određivanje razine čvora. Svakoj vrijednosti ćemo redom dodijeliti jedan nasumičan broj (nema veze što ih možemo imati viška).

Vrijednost	16	8	3	13	1	15	2	4
Nasumičan broj	0.81	0.99	0.69	0.91	0.68	0.42	0.09	0.18

Potom je potrebno odrediti razinu čvorova, koristeći jednadžbu za vjerojatnost postignuća k-tog stupnja čvora. Trebamo interval $\langle 0,1 \rangle$ podijeliti na određen broj podintervala; ovisno u koji interval nasumičan broj neke vrijednosti padne, čvor te vrijednosti će pripasti toj razini.

Broj intervala će biti h:

$$h = \lfloor 1 + \log_2 8 \rfloor = 4$$

Podijelimo sada interval na podintervale tako da odredimo granice koristeći:

$$P(k) = p^{k-1}(1 - p)$$

$$P(1) = p^0(1 - p) = 0.5$$

$$P(2) = p^1(1 - p) = 0.25$$

$$P(3) = p^2(1 - p) = 0.125$$

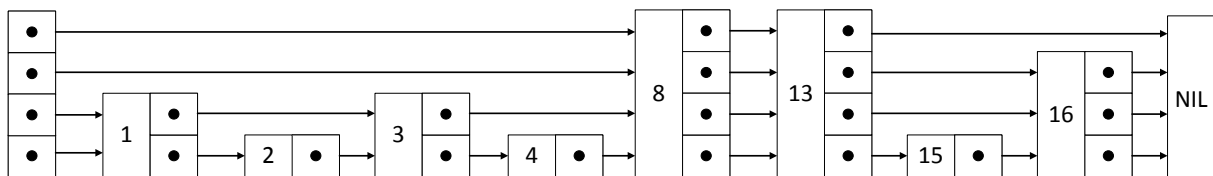
Dakle naši intervali će biti:

$$\langle 0, 0.5 \rangle, [0.5, 0.75], [0.75, 0.875], [0.875, 1)$$

I oni redom opisuju vjerojatnost da neka razina bude dodijeljena nekom čvoru (1., 2., 3., 4. razina).

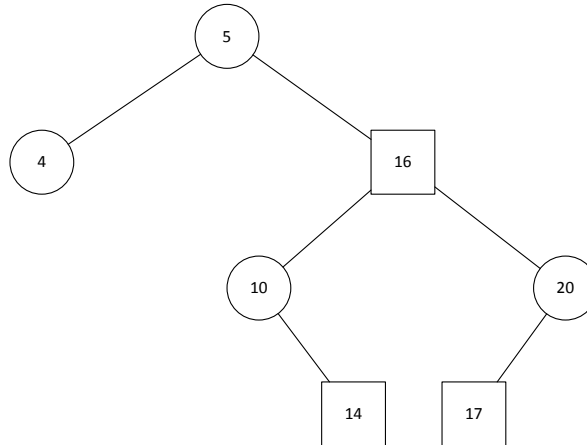
Vrijednost	16	8	3	13	1	15	2	4
Nasumičan broj	0.81	0.99	0.69	0.91	0.68	0.42	0.09	0.18
Razina	3.	4.	2.	4.	2.	1.	1.	1.

Ti čvorovi ulaze u listu sortiranim redoslijedom, te sortirana lista u konačnici izgleda:



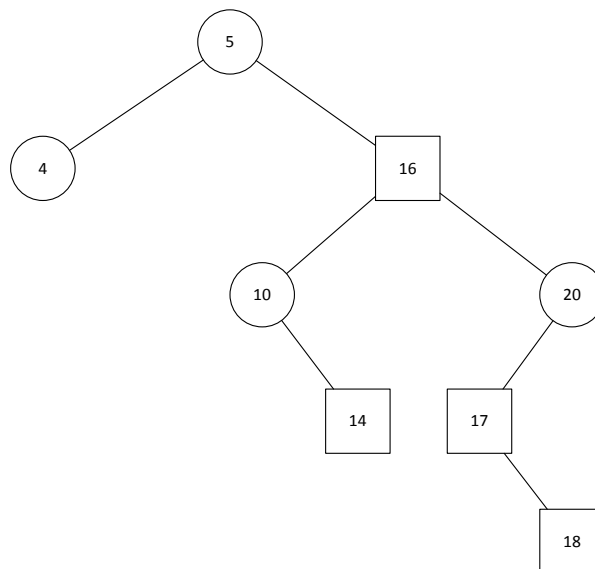
2. zadatak

(5) Dodajte 18 pa 21 u crveno-crno stablo na slici. Crveni čvorovi su kvadratični, a crni okrugli. Takvo označavanje primijenite i u Vašim rješenjima.



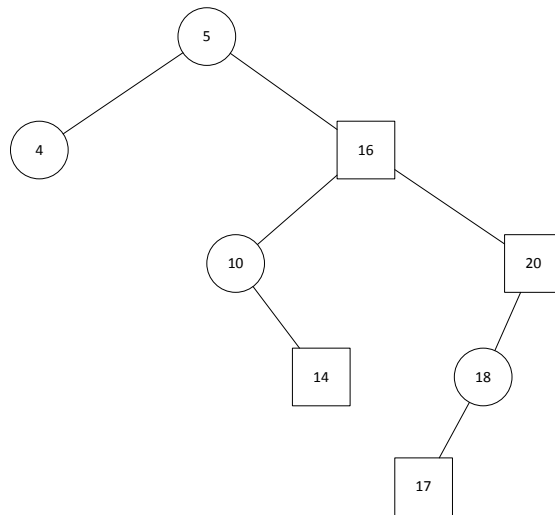
Dodajemo 18:

Pritom pazimo na osnovno pravilo Crveno-crnih stabala – novi čvorovi se dodaju kao crveni.

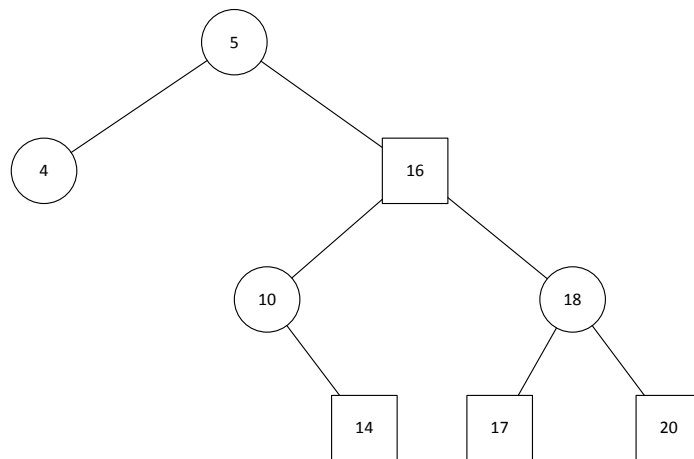


No, sada smo prekršili pravilo koje govori o tome kako se crveni čvorovi ne smiju dodirivati (ne smiju imati direktan odnos roditelj-dijete). Potrebno je izvršiti balansiranje. U ovom slučaju ujak ne postoji, odnosno gledamo NULL list gdje bi se on trebao nalaziti. NULL listovi su uvijek CRNI. U službenom podsjetniku gledamo stablo odluke i prelazimo lijevo podstablo i odabiremo slučaj raspodjele čvorova (N – novi čvor, P – roditelj, G – djed). Imamo LR slučaj i zaključujemo da je potrebno napraviti RL rotaciju (prvo na desno, pa na lijevo). Prije rotacije N-u i G-u je potrebno međusobno zamijeniti boju.

Nakon prve rotacije i zamjene boja imati ćemo:

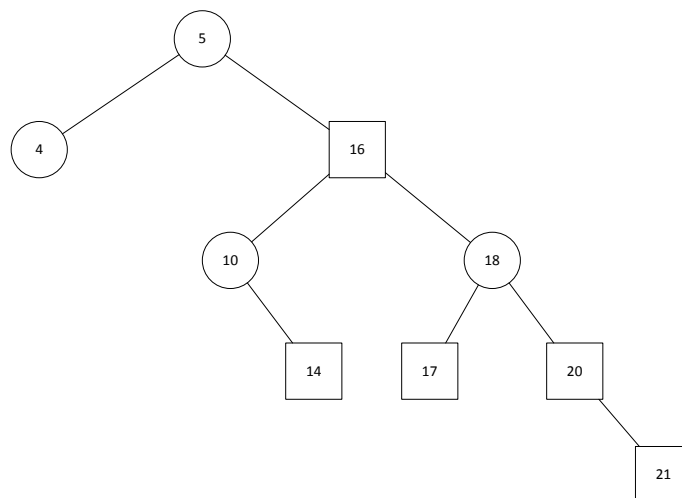


Nakon druge rotacije stablo će ponovno poštivati sva pravila:

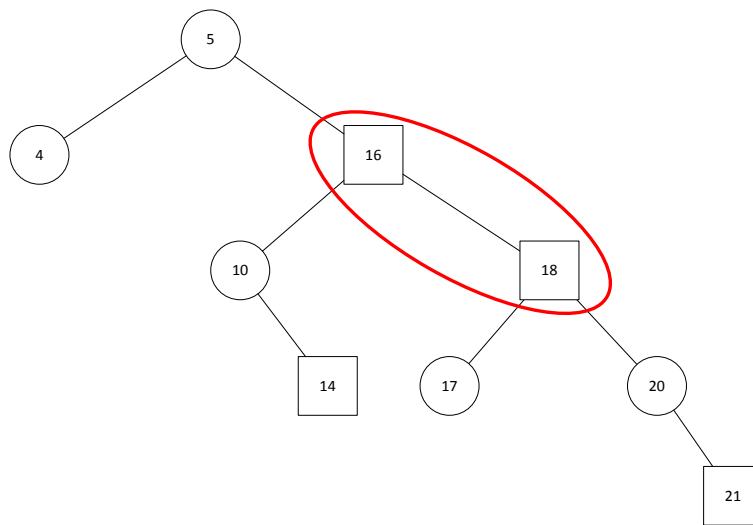


Dodajemo 21:

21 će prvotno naći mjesto kao krajnje desni list stabla.

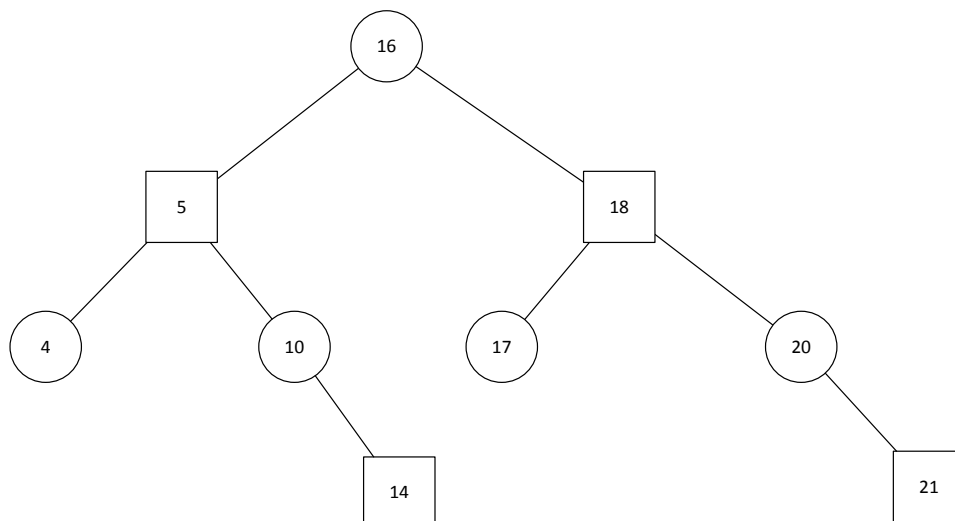


No, opet imamo dodirivanje crvenih čvorova (20, 21). U ovom slučaju ujak postoji i još je k tome crven, što olakšava postupak balansiranja. Sada je samo potrebno promijeniti boje P, G i U čvorovima (crni će postati crveni i obratno).



Algoritam bi u ovom koraku završio da se nije ponovno dogodilo dodirivanje crvenih čvorova. Moramo nastaviti rotacije sve dok stablo ponovno ne poštuje pravila crveno-crni stabala. Čvor 18 nam je sad čvor N. Čvor 4 je sada ujak i crne je boje, pa ponovno u službenom podsjetniku možemo iščitati slijedeću rotaciju i izmjenu boja.

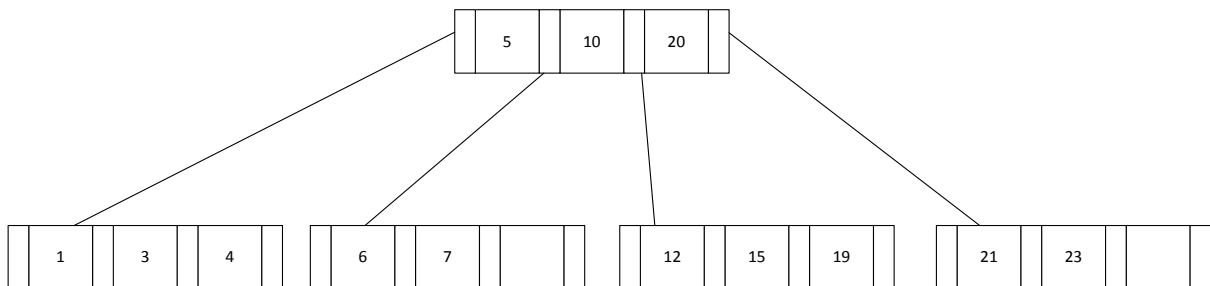
Uzimamo u obzir samo N, P i G, te zaključujemo da imamo jednostavni R slučaj. U jednostrukim rotacijama P i G mijenjaju boju. U dvostrukim rotacijama je između rotacija isto moguće zamijeniti P i G boju (ovisi kako pamtim postupak). Konačni oblik stabla će biti.



3. zadatak

(8) Napišite pseudokod funkcije za pretraživanje B-stabla. Sami odlučite što će biti argumenti, pod pretpostavkom da su čvorovi uobičajene strukture (slične onoj navedenoj na predavanjima i u mjerodavnoj literaturi). Radi jasnoće, poželjno je navesti strukturu čvora, s naznakama značenja članskih varijabli. Pri tome je dovoljan i samo popis bez ikakvih sintaktičkih elemenata.

Kako nije naveden red B-stabla (m), a to označava koliko će čvor maksimalno imati djece (i da će držati $m-1$ vrijednosti), traži se postupak za generično B-stablo. Neko B-stablo reda 4 može izgledati ovako:



Struktura i algoritam bi izgledali ovako:

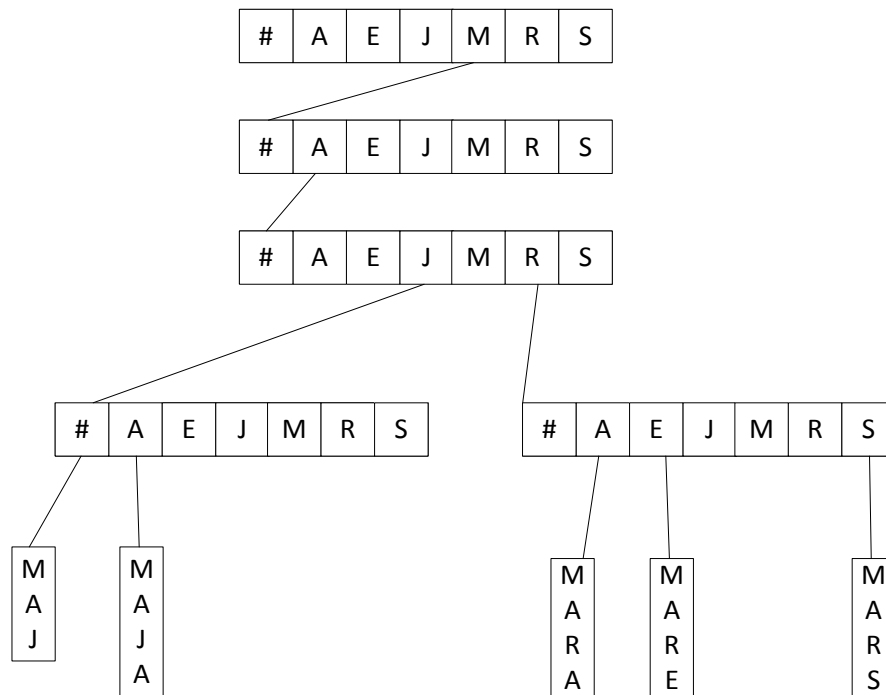
```
typedef struct Node {
    var key[m-1];
    struct Node pointers[m];
} node;
```

```
struct Node SearchBTree (var key, struct Node node){
    if (node != NULL)
    {
        for (i=1; i<=node->keyNum && node->keys[i]<key; ++i)
        {
            if (i>node->keyNum || node->keys[i]>key)
                return SearchBTree (key, node->pointers[i]);
            else
                return node;
        }
    }
    else
    {
        return null;
    }
}
```

4. zadatak

(4) Skicirajte strukturu Trie koja sadrži riječi: MARS, MARA, MARE, MAJ i MAJA.

Trie struktura je nebalansirano stablo. Čvor sadrži sve elementarne vrijednosti iz skupa pretraživih riječi. U zadatku je zadano 5 riječi. Tih 5 riječi sadrži 6 različitih znakova (A, E, J, M, R, S), stoga će čvorovi ovog Trie imati 6 referenci na slijedeći istovrsni čvor i jednu referencu na trenutnu riječ (#). Također je praktično znakove poredati abecedno, jer je tako lakše implementirati pretragu. Kako se spuštamo niz stablo, tako odabiremo slijedeći znak koji se nalazi u riječi. Listovi ovog stabla su unesene riječi.



Specifičan slučaj se dogodio kod riječi MAJ, koja je očito trebala biti samo običan list. No, kako postoje riječi kojima je MAJ prefiks, MAJ smo prebacili u posebnu referencu čvora. Time smo naznačili da se ta riječ nalazi na toj razini, te je i ona pretraživa.