



DUOMENŲ STRUKTŪROS IR ALGORITMAI

Hierarchinės duomenų struktūros :

AVL medis, Bajero medis, piramidė (heap)

Praeitios paskaitos santrauka

- Hierarchinės duomenų struktūros apibrėžimas
- Dvejetainis medžiai
 - Apibrėžimas
 - Pilnas medis, subalansuotas medis
 - Viršūnių apėjimo algoritmai (tiesioginis, atvirkštinis, vidinis)
- Dvejetainis paieškos medis

AVL medis

AVL medis - tai iš dalies subalansuotas dvejetainis paieškos medis, kurio bet kurios viršūnės kairiojo ir dešiniojo pomedžio aukščiai gali skirtis ne daugiau nei vienetu. Jis vadinamas AVL medžiu pagal jį sukūrusių matematikų G. Adelson–Velskio ir E. Landi pavardes. Medžio principai buvo paskelbti straipsnyje "An algorithm for the organization of information" 1962.

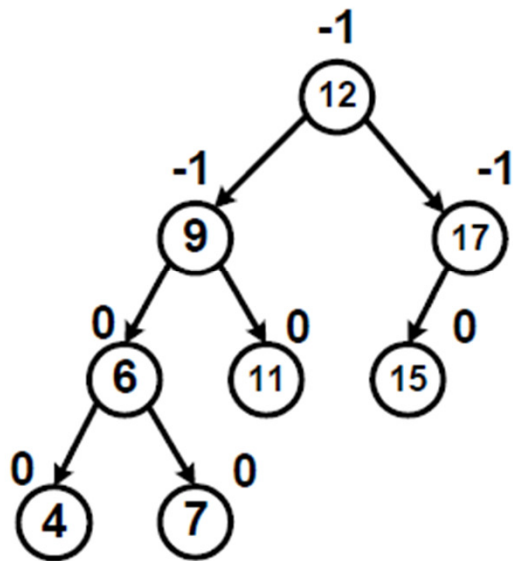
Pagrindinė AVL medžio savybė:

- Kiekvienos viršūnės kairiojo ir dešiniojo pomedžio aukščiai gali skirtis ne daugiau nei 1.

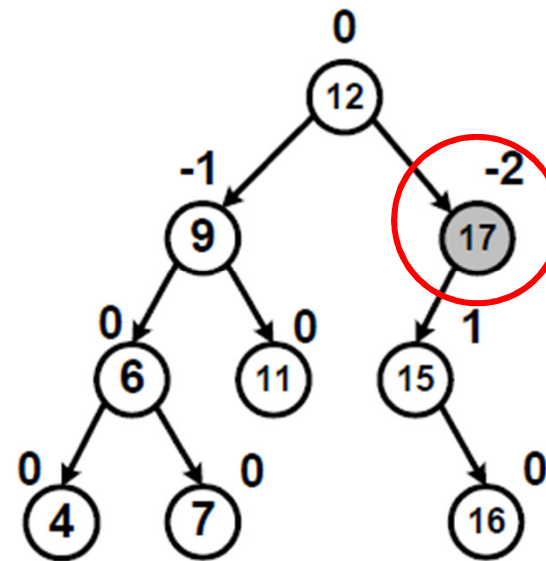
Skirtumas tarp pomedžių aukščių vadinamas balansiniu faktoriumi.

AVL medžio bet kurios viršūnės balansinis faktorius gali būti 1, 0 arba -1.

AVL medis



AVL medis

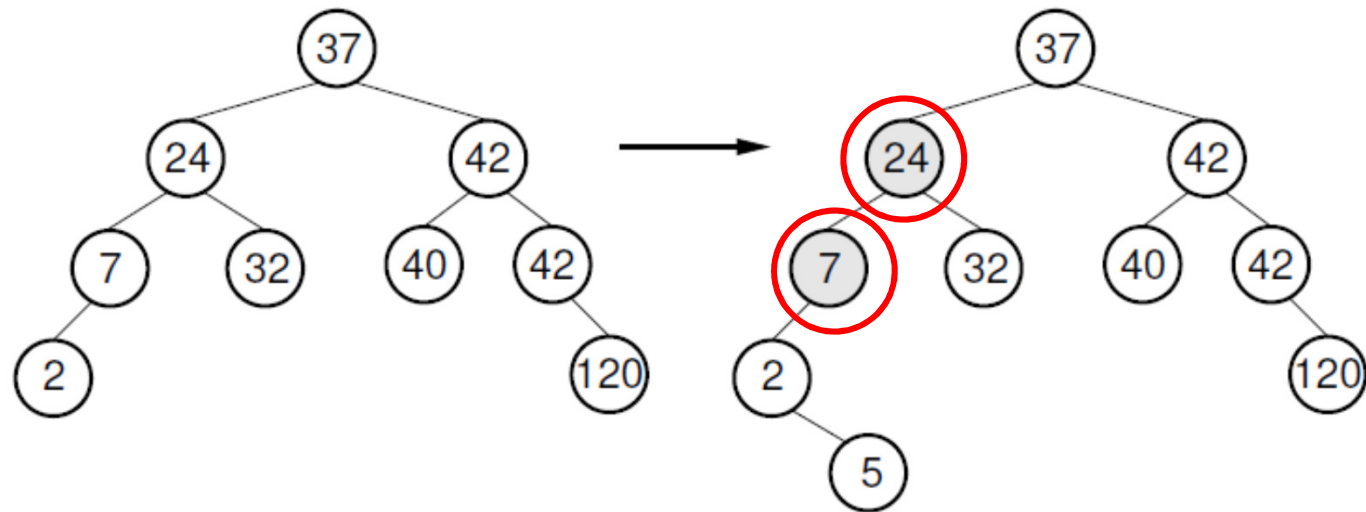


Ne AVL medis

AVL medžio elementas

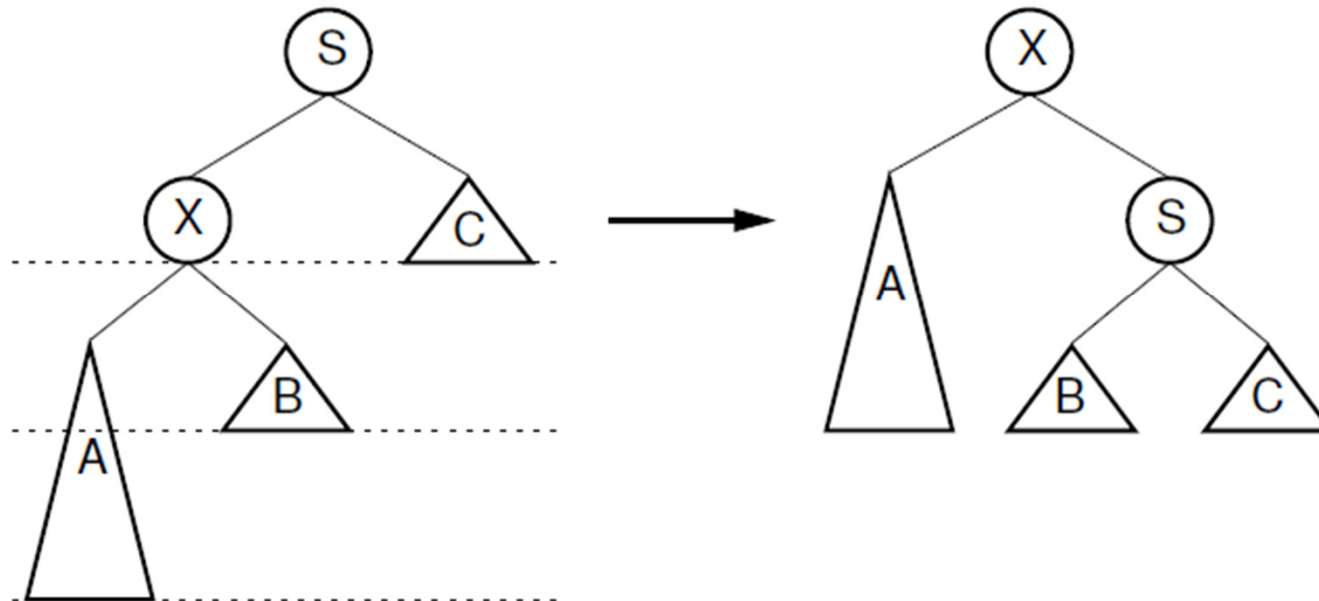
```
struct AVLnode
{
    int data;
    AVLnode* left;
    AVLnode* right;
    int factor;           // viršūnės balansinis faktorius
}
```

Priedamas naujas elementas



Priedama viršūnė su reikšme 5. Viršūnės pridėjimo algoritmas toks pat kaip paieškos medžio. Jei pridėjus elementą medis išsibalansuoja t.y. balansiniai faktoriai tampa didesni nei 1 ar -1, būtina atlikti medžio viršūnių posūkius.

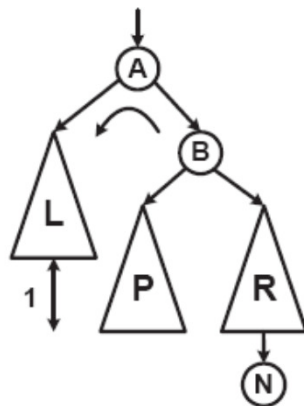
Posūkis (į dešinę)



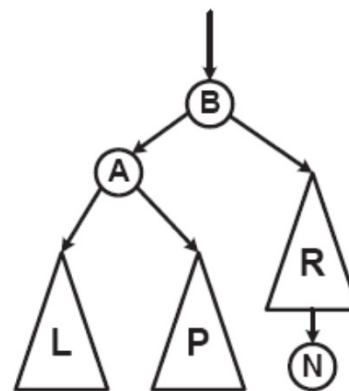
Sakykime, jog turime nesubalansuotą medį.

Norint jį subalansuoti, reikia atlikti posūkį apie S viršūnę. Atlikus posūkį, medis subalansuojamas t.y. abu šaknies pomedžiai tampa vienodo aukščio.

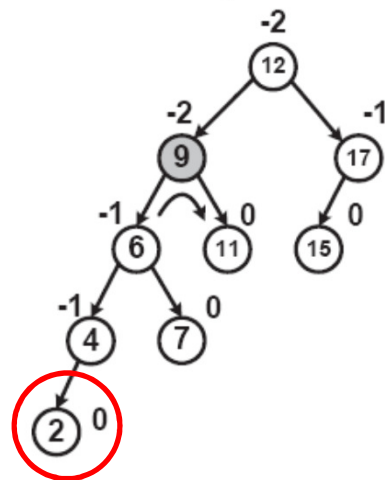
Posūkis ģ kairē



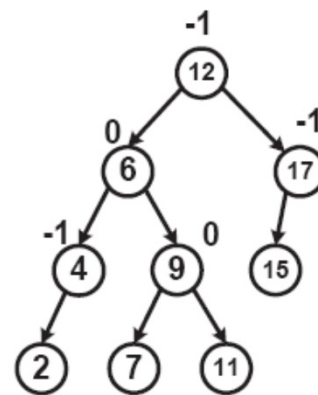
a)



b)

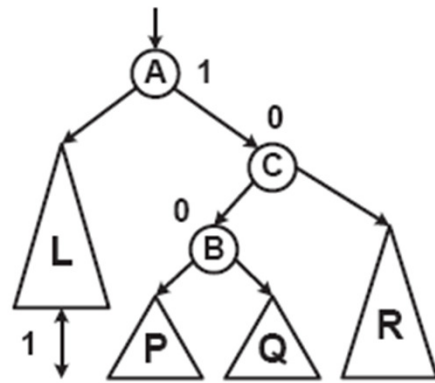


AVL medis ģterpus
naujā viršūņē 2

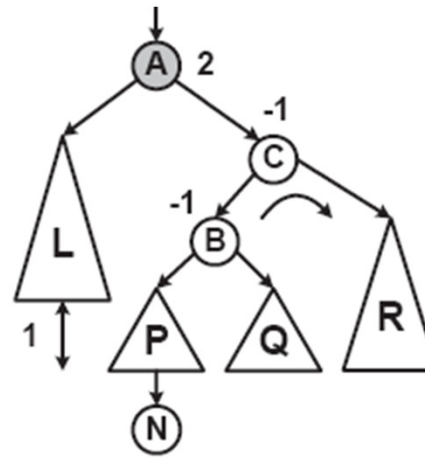


AVL medis po pasūkimo

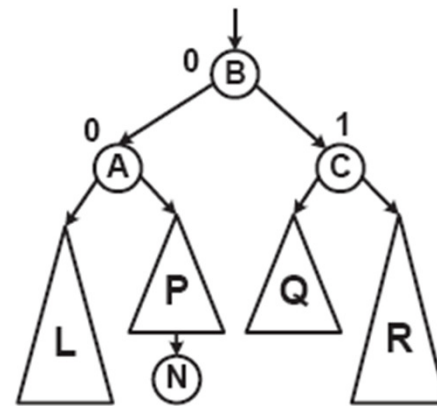
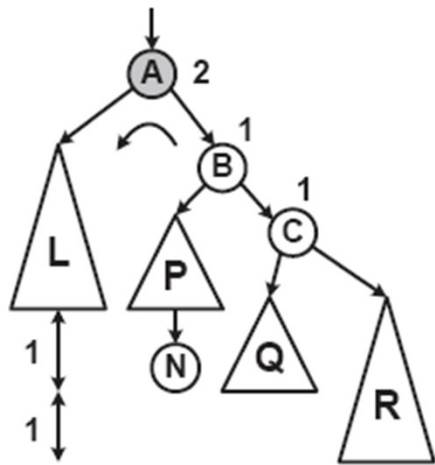
Du posūķiai



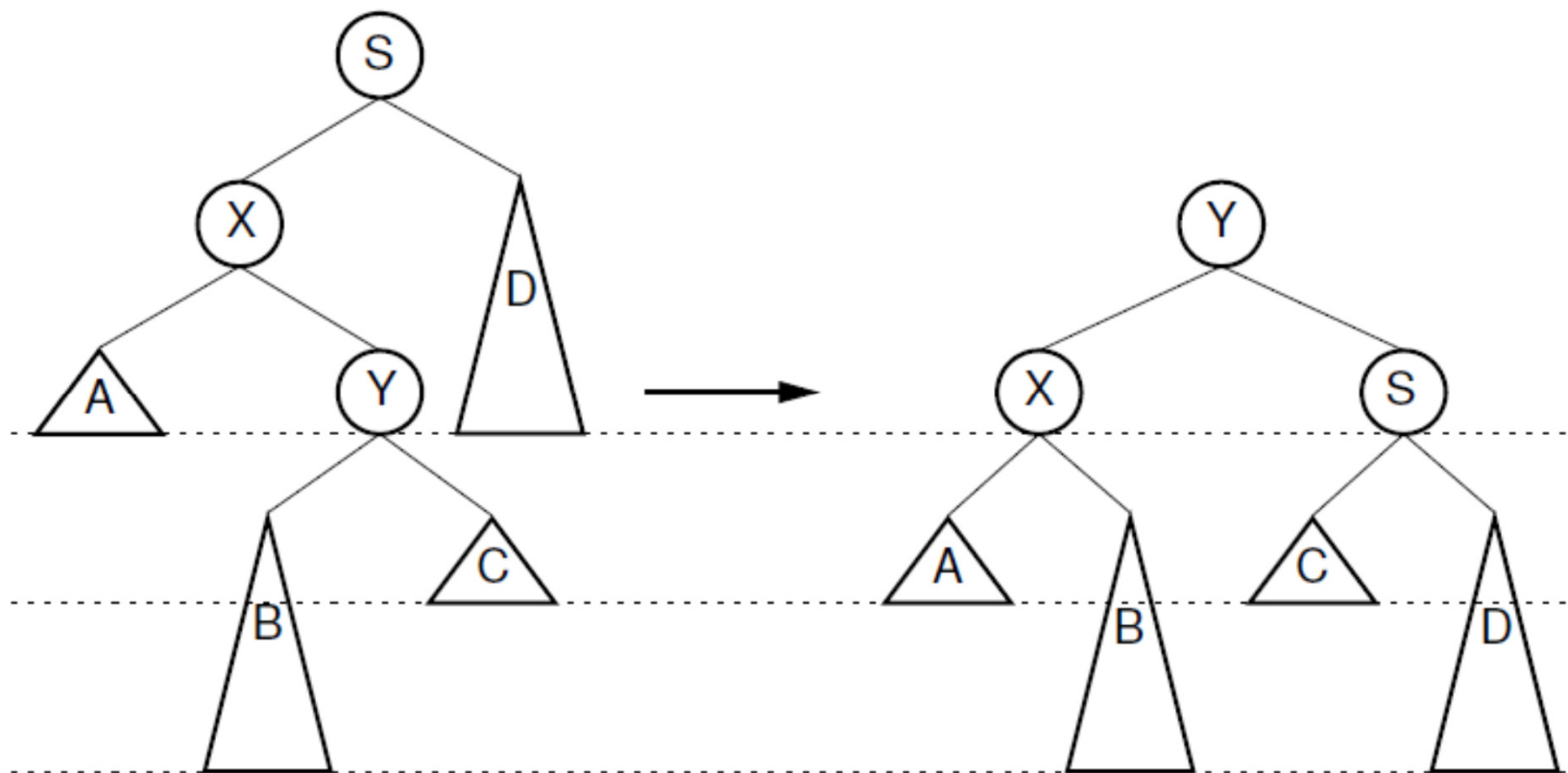
a)



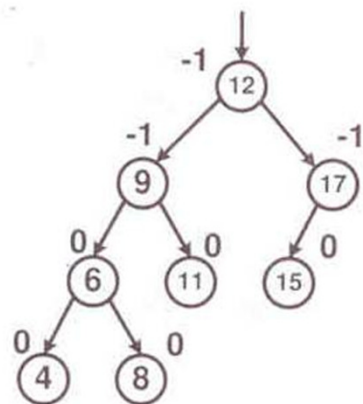
b)



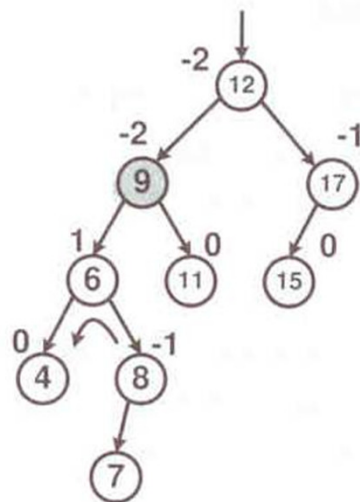
Dvigubas posūkis



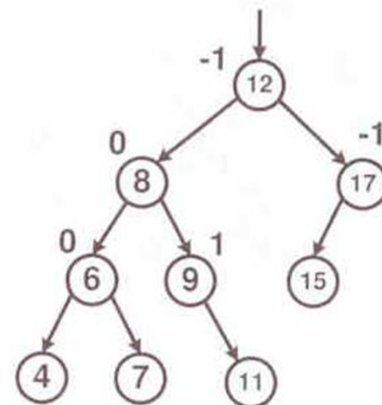
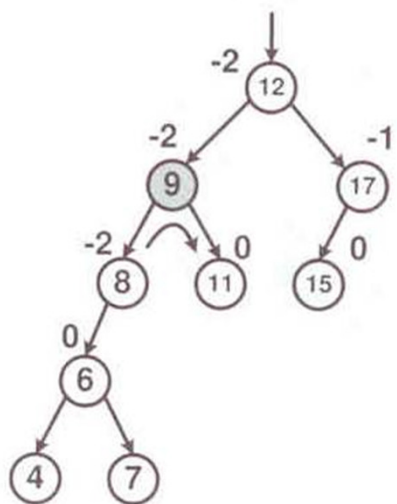
Dvigubas posūkis



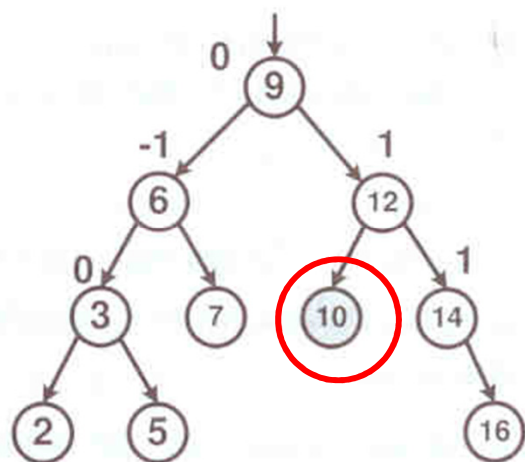
a)



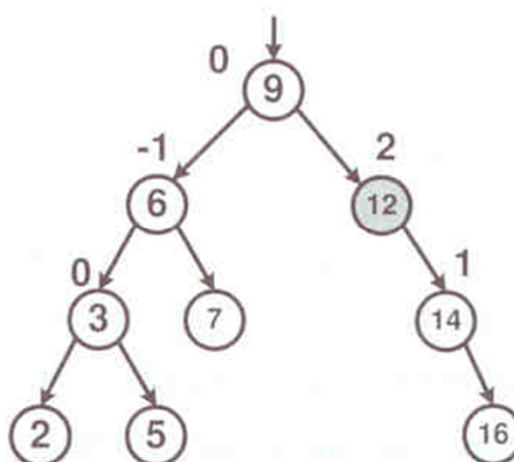
b)



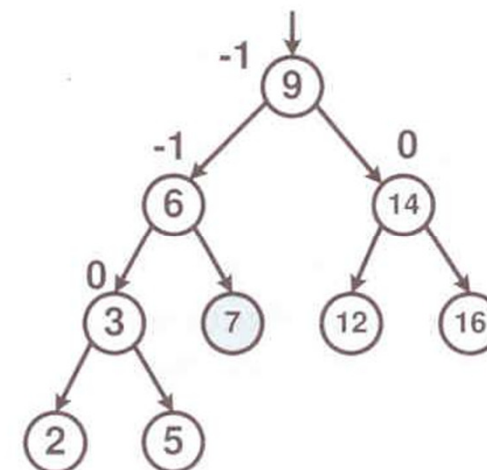
Viršūnės šalinimas



Pradinis AVL medis



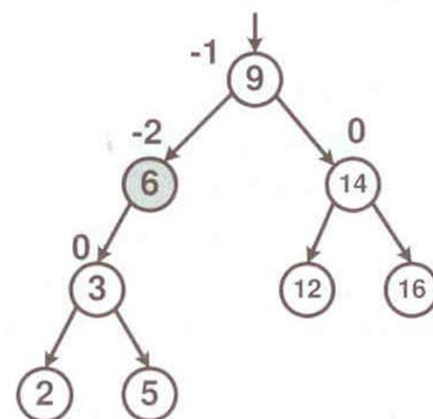
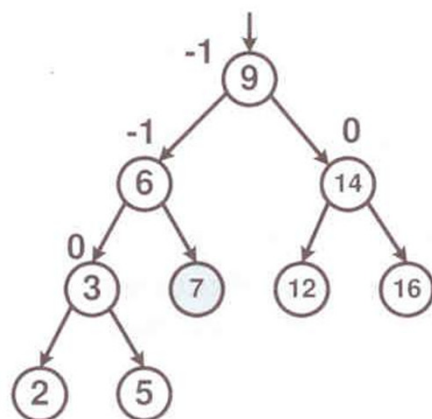
Pašalintas lapas 10



Sutvarkytas AVL medis

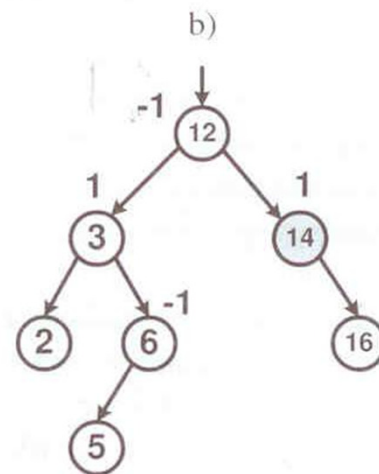
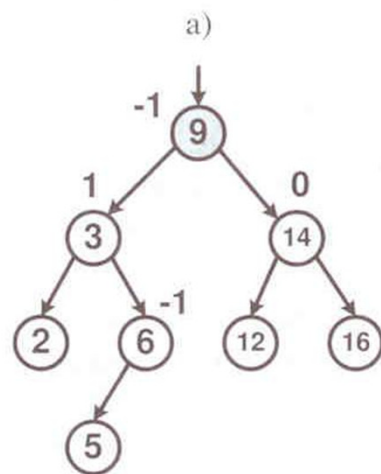
Viršūnēs šalinimas

AVL medis



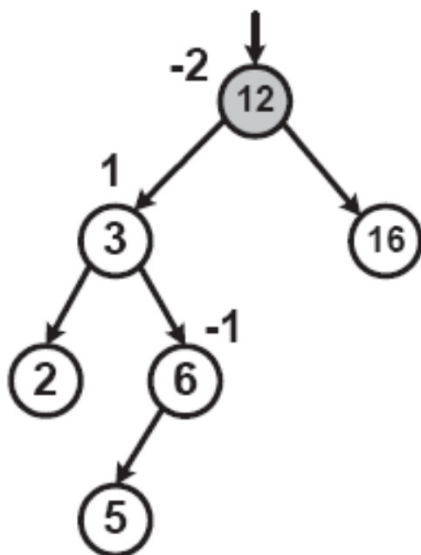
Pašalintas lapas 7

Dvīgubas posūkis

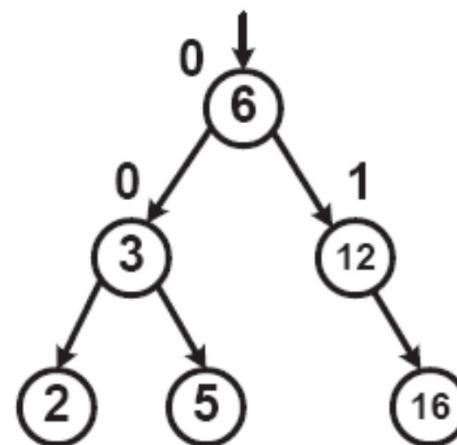


Pašalinta viršūnē 9

Viršūnės šalinimas



Pašalinta viršūnė 14



Sutvarkytas AVL medis

Bajero medžis

- **Bajero arba trumpiau B-medis** – tai apibendrintas dvejetainis paieškos medis, kurio viršūnės gali turėti daugiau nei du vaikus.
- **m** eilės B-medis gali turėti iki **m** vaikų.
- **Savybės:**
 - Visose viršūnėse (ne lapuose) elementų (raktų) skaičius vienetu mažesnis nei vaikų skaičius.
 - Visi lapai yra viename lygmenyje.
 - Visos viršūnės išskyrus šaknį turi ne mažiau nei **m/2** vaikų.
 - Šaknis yra arba lapas, arba turi nuo 2 iki m vaikų.
 - Viršūnės lapai turi ne daugiau nei **m – 1** raktą.

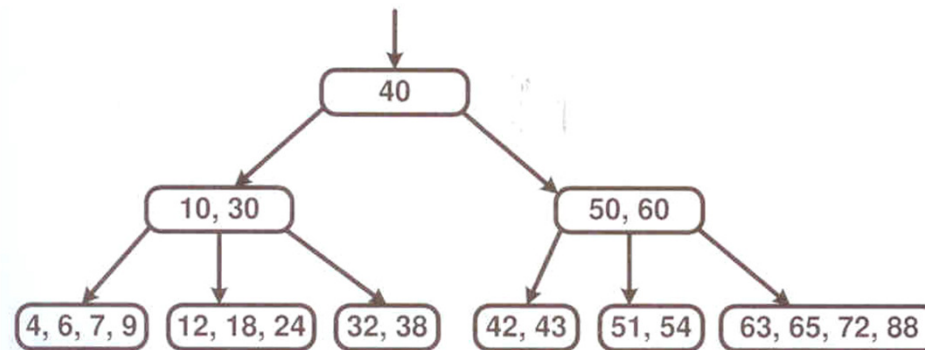
B-medis

B-medyje laikomi duomenys yra išrūšiuoti, o paieška, duomenų pasiekimas, įterpimas ir trynimasis atliekamas pagal logaritminę skalę.

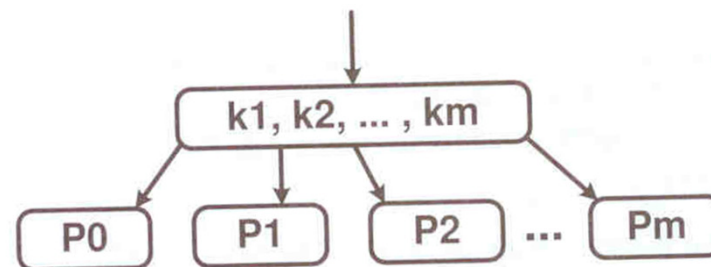
B-medis yra skirtas sistemoms, kuriose reikia atlikti greitus paieškos, skaitymo-rašymo veiksmus su dideliais duomenų blokais. Dažniausiai tokia duomenų struktūra naudojama duomenų bazių valdymo sistemose arba failų sistemose.

B-medžiai visada būna subalansuoti medžiai.

B-medžio pavyzdys



5 eilės B-medis

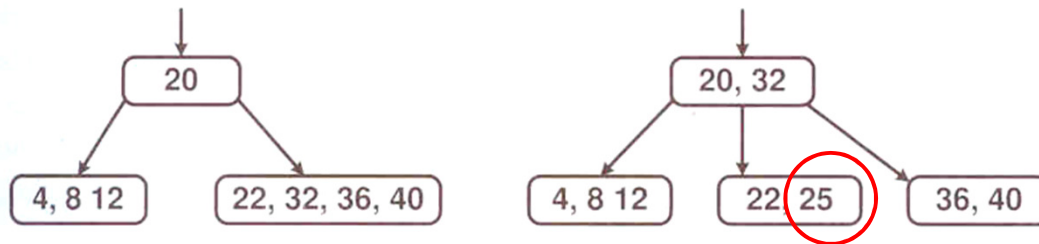


B-medžio pagrindinė savybė:

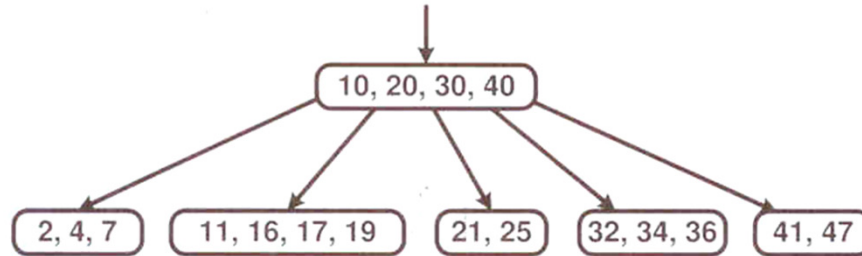
$P_0 < k_1 < P_1 < k_2 < P_2 < \dots < P_{m-1} < k_m < P_m$, kur $m \leq 2N$

Naujo rakto pridėjimas

Pradinis B-medis

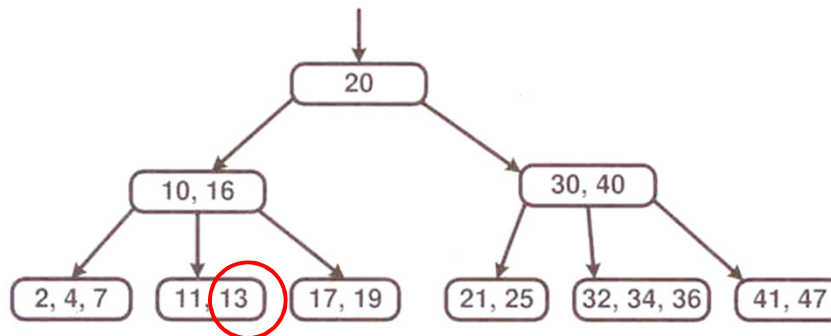


Modifikuotas B-medis



a)

Modifikuotas B-medis



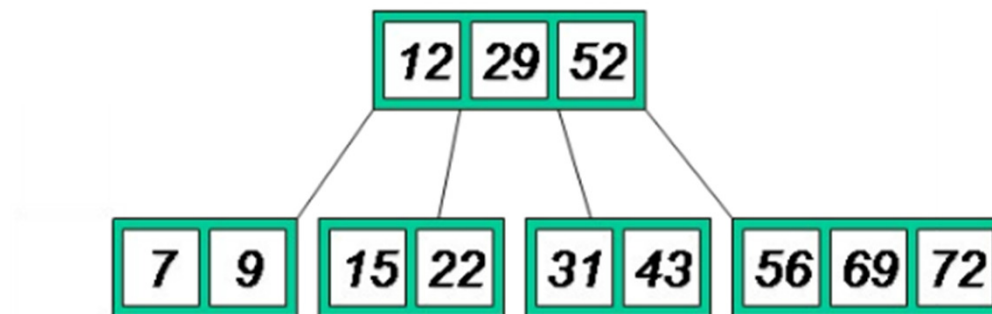
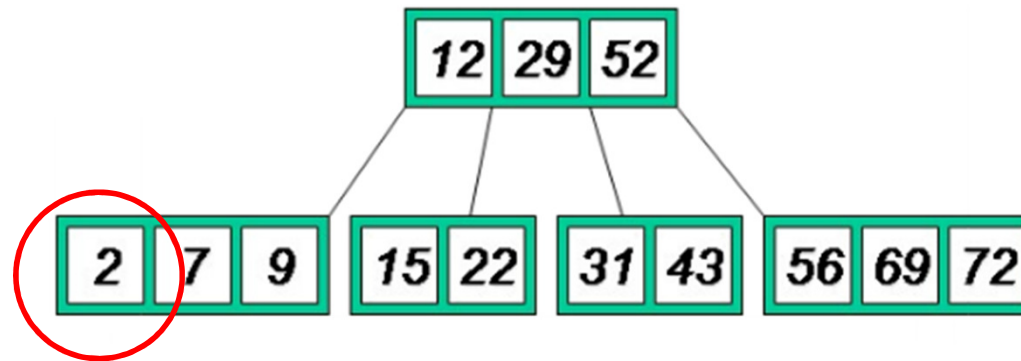
Naujo rakto pridėjimas

- Bandome pridėti raktą į viršūnę lapą.
- Jei lapas pasidaro per didelis (per daug raktų), lapą skeliame į dvi dalis ir vidurinį raktą perkeliame pas tėvą.
- Jei tėvas tampa per didelis, jis skeliamas pusiau ir vidurinis mazgas perkeliamas pas aukštesnį tėvą.
- Veiksmai kartojamas, kol pasiekiamas šaknis.
- Jei šaknyje pasidaro per daug raktų, ji skeliama į dvi viršūnes ir formuojama nauja šaknis.

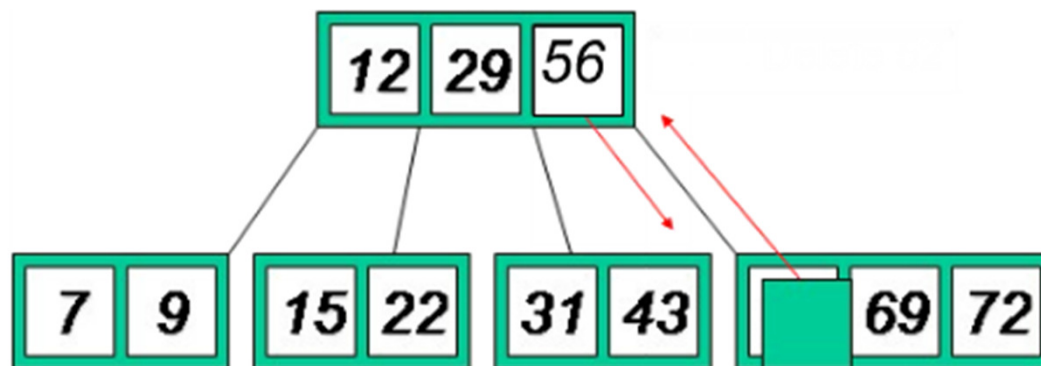
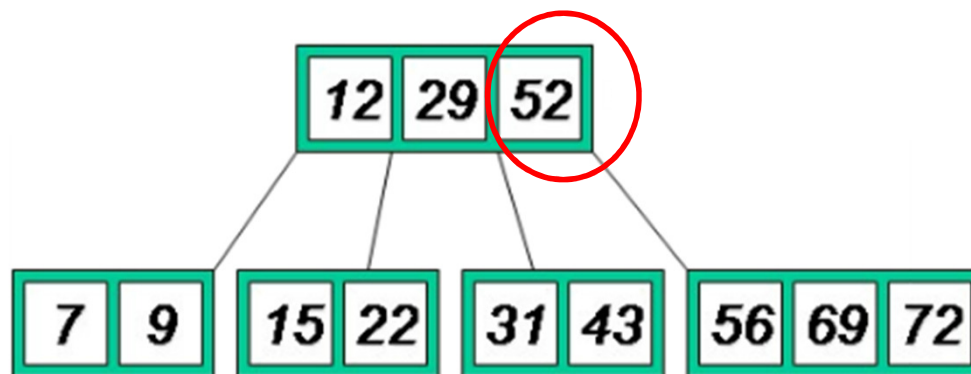
Rakto trynimas

- Jei trinamas raktas yra viršūnėje lape ir jį pašalinus viršūnėje raktų skaičius lieka ne per mažas, tuomet raktas paprasčiausiai pašalinamas
- Jei trinamas raktas yra paprastoje viršūnėje, tuomet pašalinus raktą į jo vietą perkeliamas raktas iš viršūnės (vaiko).

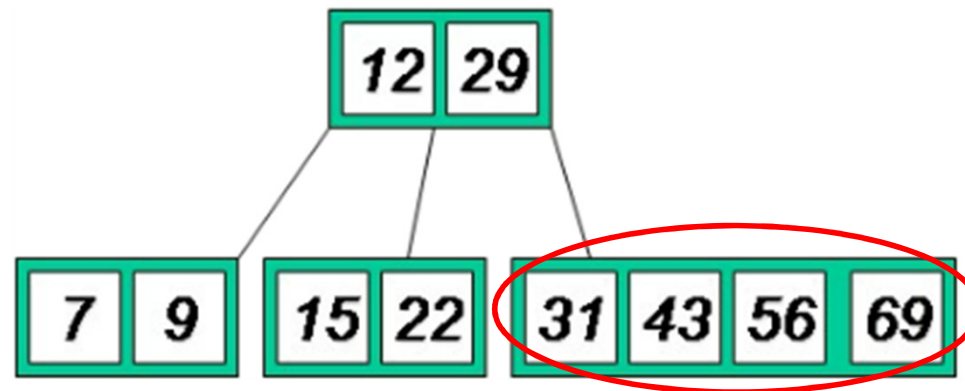
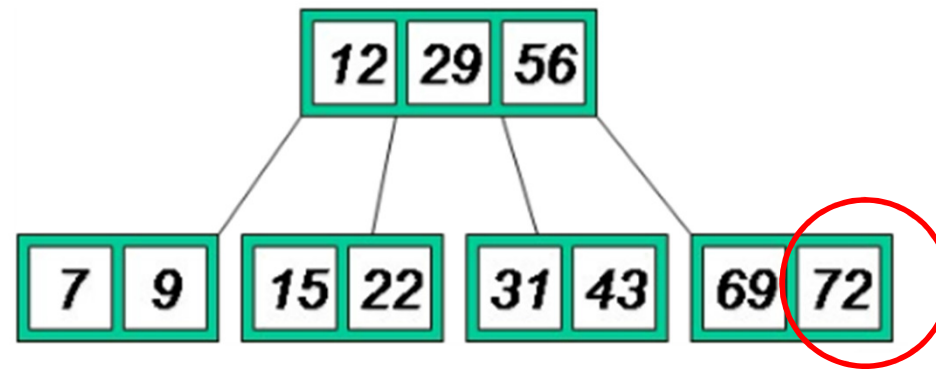
Lapo rakto trinimas



Viršūnēs rakto trynimas



Lapo rakto trynimas su apjungimu

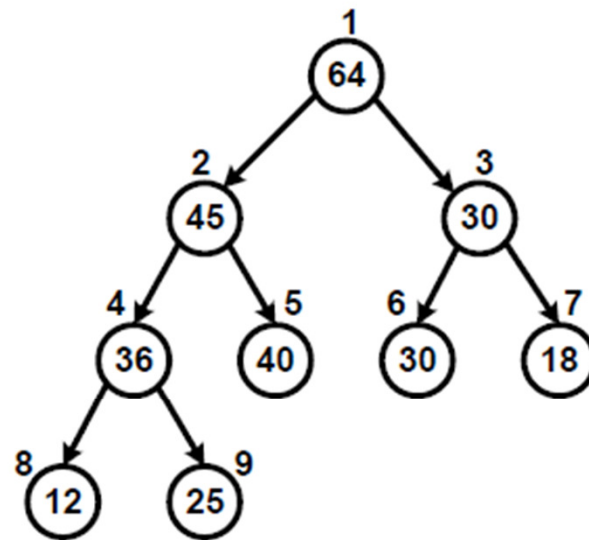


Dvejetainė piramidė

Dvejetainė piramidė tai dvejetainis medis, kuriam būdingos šios dvi savybės:

- Piramidė yra pilnas, subalansuotas medis t.y. pirmiausia užpildoma medžio šaknis, po to pirmojo lygmens viršūnės, antrojo lygmens viršūnės ir t.t. Kiekvieną lygmenį užpildome iš kairės į dešinę.
- Piramidė yra sutvarkyta taip, kad kiekvienos viršūnės vaikai yra nedidesni už pačią viršūnę. Didžiausia viršūnė - šaknis.

Pavyzdys



a)

1	2	3	4	5	6	7	8	9
64	45	30	36	40	30	18	12	25

Dvejetainė piramidė

Kadangi piramidė yra pilnas dvejetainis medis, tai jos viršūnes patogiu saugoti masyve.

Tada i -tosios viršūnės a_i vaikai yra a_{2i} , a_{2i+1} masyvo elementai.

Lengvai randame kiekvienos piramidės viršūnės a_i tėvinę viršūnę: ji saugoma ($j = i/2$) masyvo elemente a_j .

Piramidės formavimas

Tarkime turime seką $e_1 \dots e_n$ iš kurių reikia sudaryti piramidės struktūrą.

1. Šiuos elementus paeiliui talpiname į A masyvą. Piramidės lapai jau sutvarkyti, o indeksai kinta nuo $(N/2 + 1)$ iki N .
2. Paeiliui imame viršūnes $N/2, \dots, 1$ ir rekursyviai tikriname piramidės sutvarkymo sąlygą. Jei ji yra pažeista, tai sukeičiame vietomis tėvinę viršūnę su didžiausiu jos vaiku.

Piramidės formavimo algoritmo sudėtingumas:

$$L(N) = N \log N, \quad S(N) = \frac{1}{2} N \log N$$