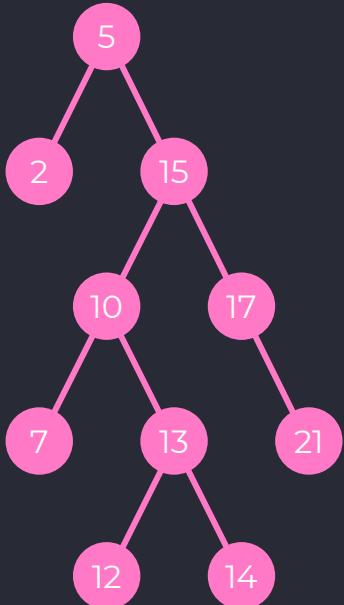


Dvejetainis paieškos medis



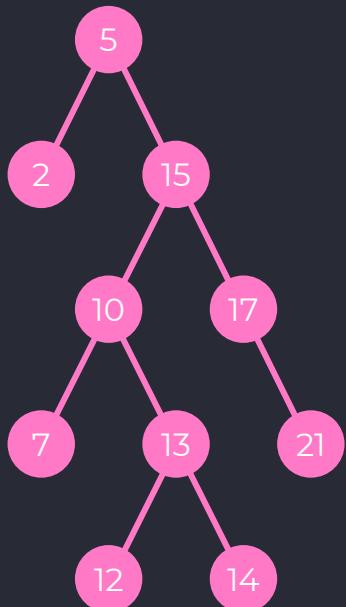
Elemento paieška

Elemento įterpimas

Elemento šalinimas

Medžio apėjimas

Savybės



Kiekviena viršūnė gali turėti iki dviejų pomedžių.

Nuo šaknies iki kiekvieno kito mazgo egzistuoja tik vienas kelias.

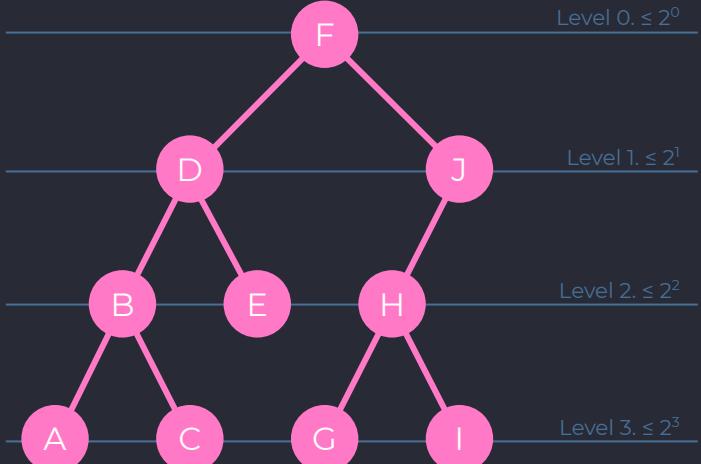
Viršūnės reikšmė yra didesnė už visas reikšmes jos kairiajame pomedyje.

Viršūnės reikšmė yra mažesnė už visas reikšmes jos dešiniajame pomedyje.

Kairysis ir dešinysis pomedžiai yra dvejetainiai paieškos medžiai.

Viršūnių reikšmės negali kartotis.

Terminologija



Nepilnas (nesubalsansuotas) medis

Pradinis mazgas vadinamas **šaknimi**

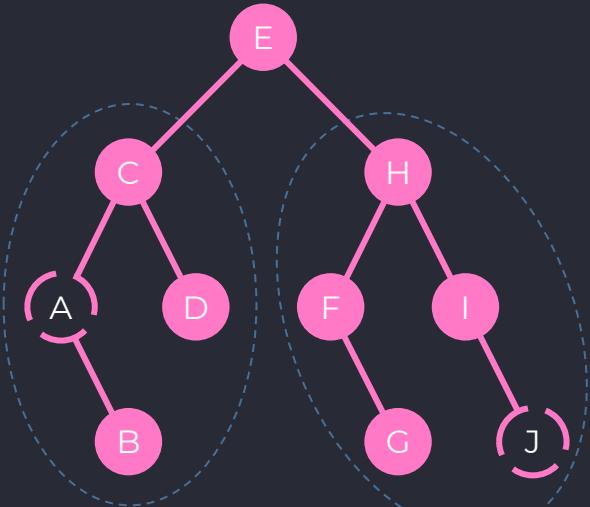
Mazgas be vaikų vadinamas **lapu**

Mazgo **lygis** (arba gylis): briaunų skaičius kelyje nuo šaknies iki to mazgo.

Medžio **aukštis h**: lygių skaičius (medžio gylis)

Pilnas medis: kiekvienas mazgas turi tiksliai du vaikus ir visi lapai yra tame pačiame lygyje.

Paieška medyje



Visos kairiajame
pomedyje esančios
reikšmės yra
mažesnės nei
šaknинio mazgo vertė

Visos dešiniajame
pomedyje esančios
reikšmės yra didesnės
nei šaknинio mazgo vertė

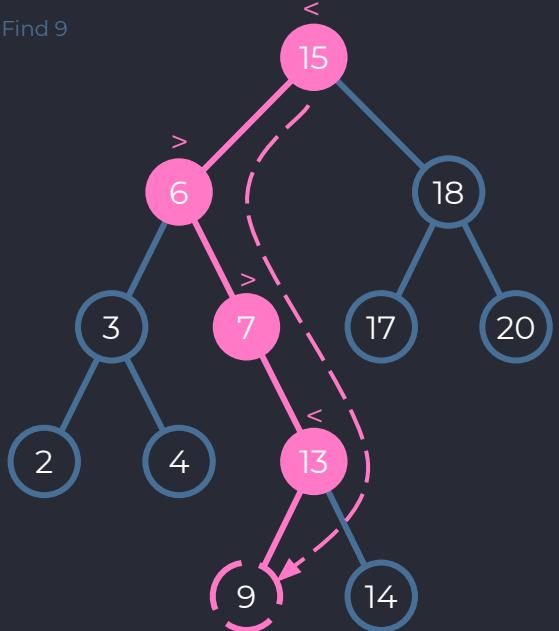
Medžio (pomedžio) šaknyje saugoma reikšmė yra
didesnė už bet kurią jo kairiojo pomedžio reikšmę ir
mažesnė už bet kurią reikšmę dešiniajame
pomedyje

Mažiausia medžio reikšmė yra saugoma
kairiajame pomedyje

Didžiausia medžio reikšmė yra saugoma
dežiniajame pomedyje

Paieška medyje

Find 9



1. Pradėkite nuo šaknies
2. Palyginkite ieškomo elemento vertę su šaknyje išsaugota verte
3. Jei reikšmės yra vienodos, tada elementas surastas; priešingu atveju, jei tai yra lapas, tada nerandama.
4. Jei ji yra mažesnė už šaknyje išsaugotą vertę, ieškokite kairiajame pomedyje
5. Jei ji yra didesnė už šaknyje išsaugotą vertę, ieškokite dešiniajame pomedyje
6. Pakartokite 2–6 veiksmus, atlikdami ankstesniame 4 arba 5 žingsnyje pasirinkto porūšio šaknį

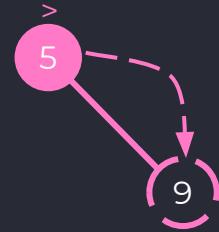
Elemento įterpimas

BST: 5, 9, 3, 4, 7, 12

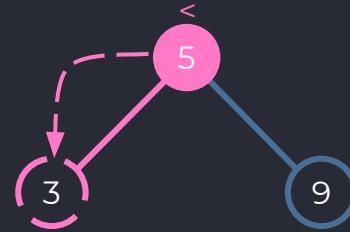
1. Insert 5



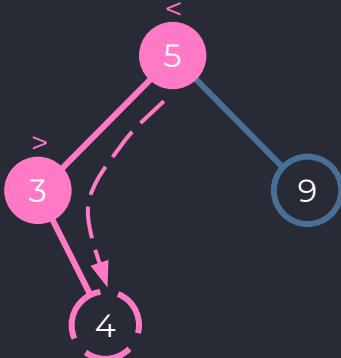
2. Insert 9



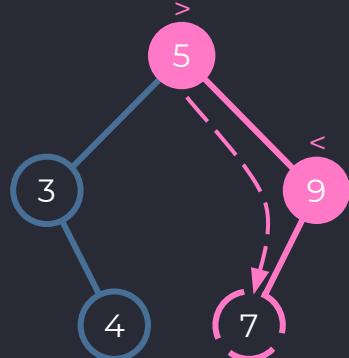
3. Insert 3



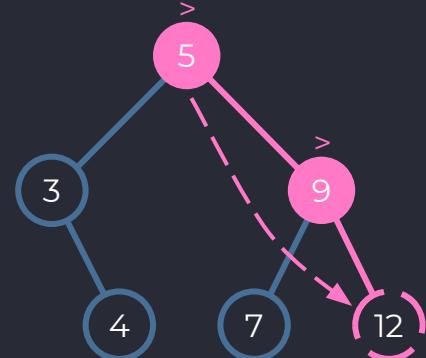
4. Insert 4



5. Insert 7



6. Insert 12

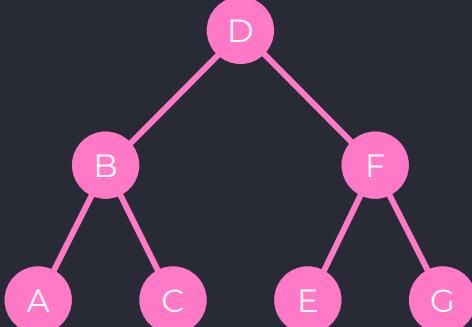


Elemento įterpimas

Elementų įterpimo eiliškumas į BST yra svarbus.

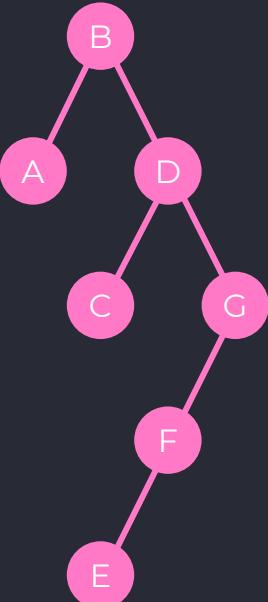
Nesubalansuoti medžiai yra nepageidaujami, nes ilgėja paieškos laikas.

Kitos duomenų struktūros, tokios kaip **raudonai-juodi** arba **AVL** medžiai, užtikrina subalansuotus medžius.



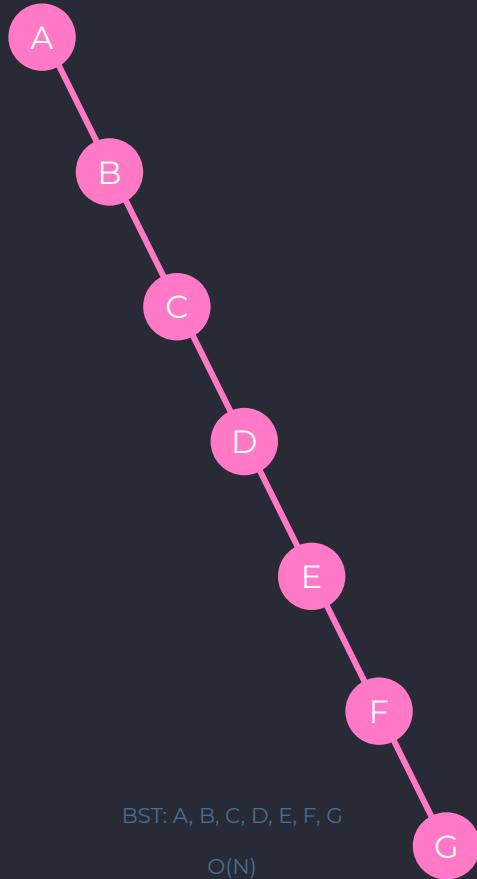
BST: D, B, F, A, C, E, G

$O(\log N)$



BST: B, A, D, C, G, F, E

$O(\log N) \dots O(N)$



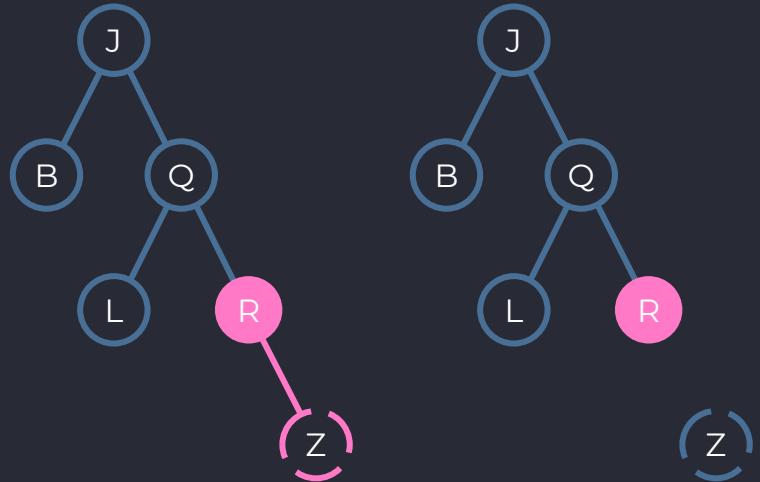
BST: A, B, C, D, E, F, G

$O(N)$

Elemento šalinimas

Pašalinti elementą Z

Lapo šalinimas



Ištrindami mazgą turime atsižvelgti į tai, kas nutiks su ištrinto mazgo vaikais. Po elemento šalinimo turi būti išlaikytos BST savybės.

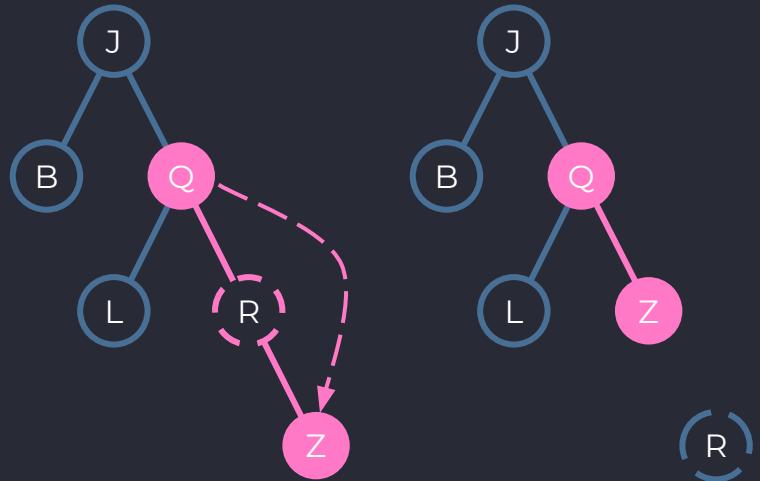
Trys atvejai:

1. Lapo šalinimas.
2. Viršūnės, turinčios vieną vaiką, šalinimas.
3. Viršūnės, turinčios du vaikus, šalinimas.

Elemento šalinimas

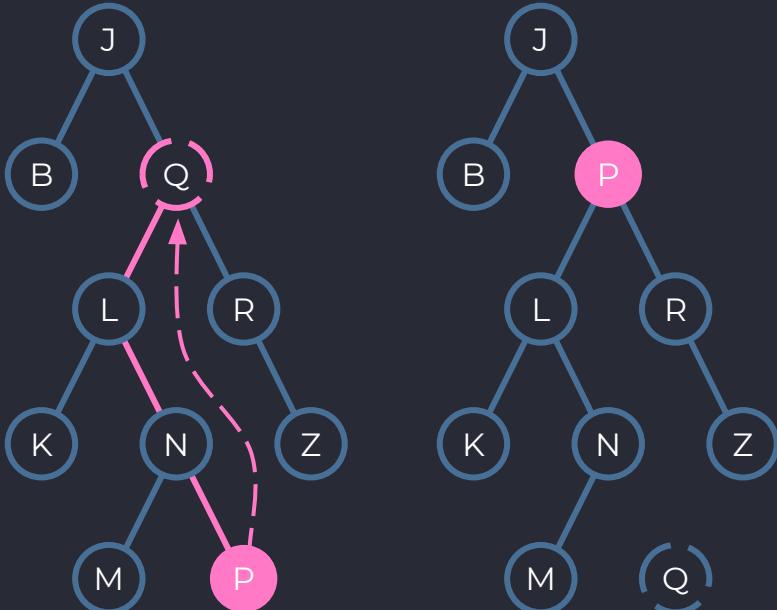
Pašalinti elementą R

Viršūnės, turinčios vieną vaiką, šalinimas.



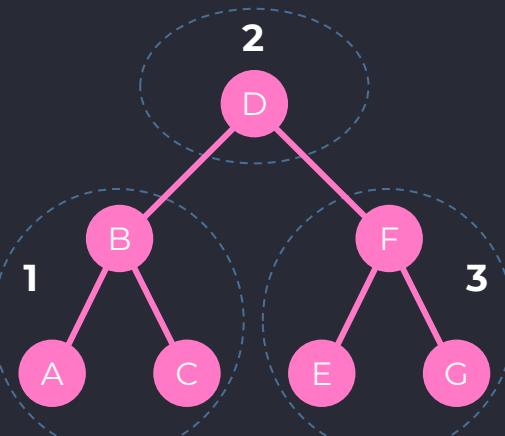
Pašalinti elementą Q

Viršūnės, turinčios du vaikus, šalinimas.



Medžio apėjimas

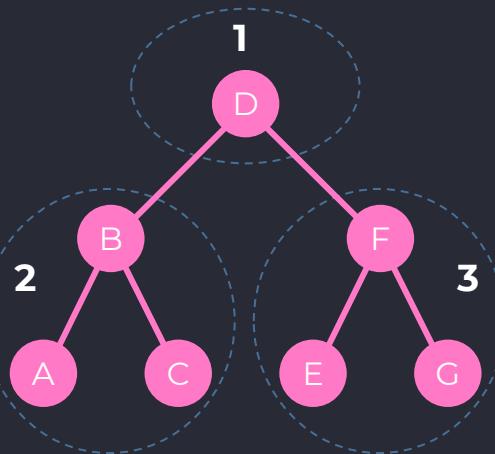
DFS: in-order



BST: A, B, C, D, E, F, G

1. Aplankykite viršūnės kairijį pomedj,
2. Aplankykite medžio šaknį,
3. Aplankykite viršūnės dešinijį pomedj.

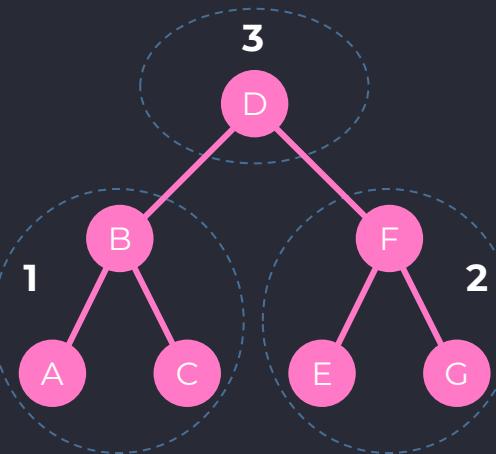
DFS: pre-order



BST: D, B, A, C, F, E, G

1. Aplankykite medžio šaknį,
2. Aplankykite viršūnės kairijį pomedj,
3. Aplankykite viršūnes dešinijį pomedj.

DFS: post-order

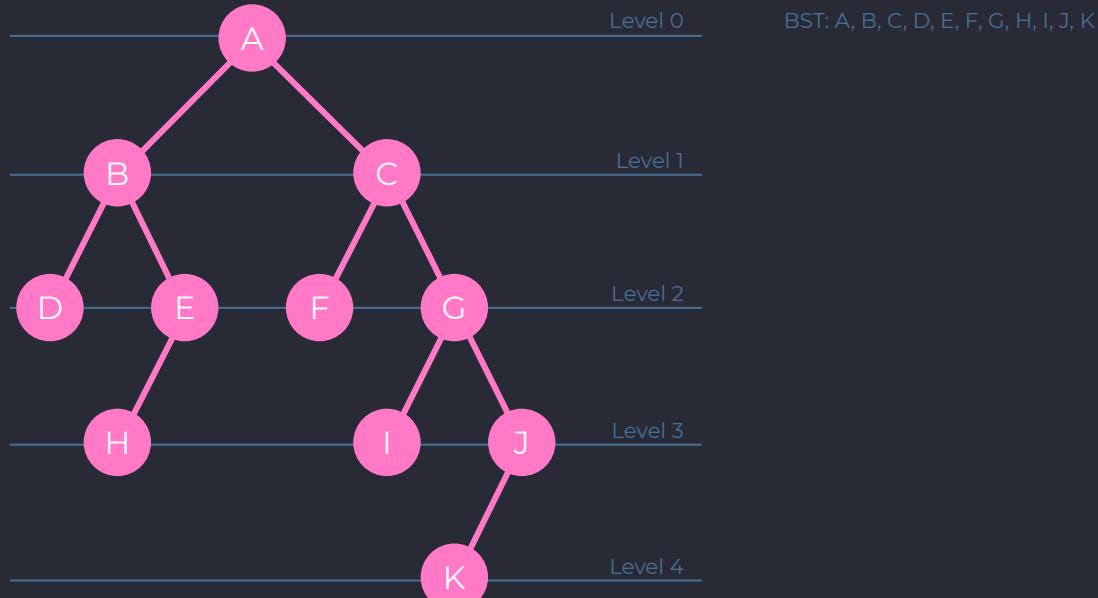


BST: A, C, B, E, G, F, D

1. Aplankykite viršūnės kairijį pomedj,
2. Aplankykite viršūnės dešinijį pomedj,
3. Aplankykite medžio šaknį.

Medžio apėjimas

BFS: level by level



Medžio apėjimas vyksta lygis - po - lygio, iš kairės į dešinę

BST: A, B, C, D, E, F, G, H, I, J, K

Ačiū už dėmesį

Klausimų bus?