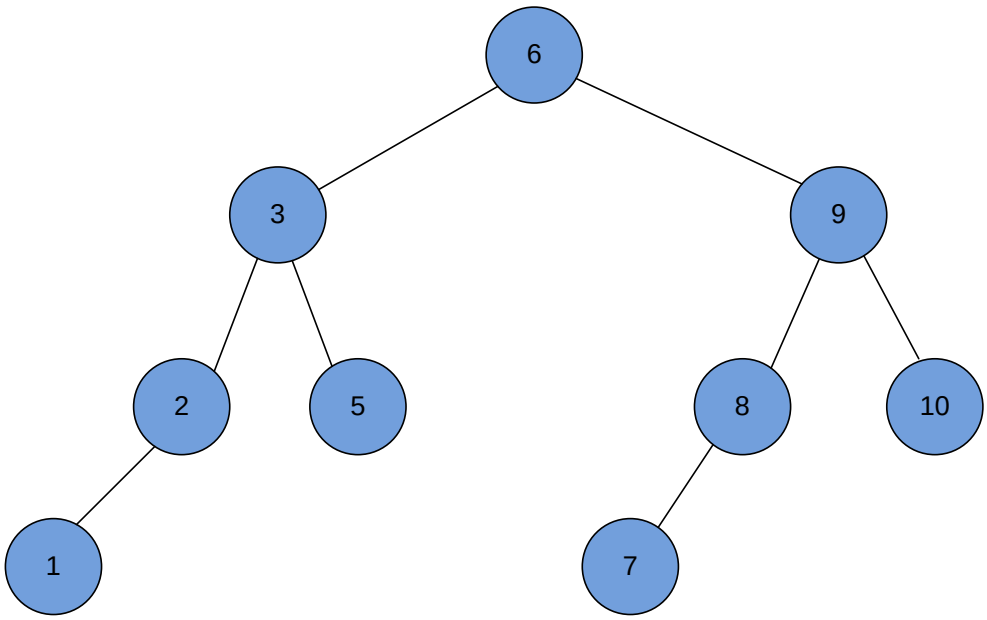
Binārais meklēšanas koks

Binārais meklēšanas koks – koks, kur katram mezglam var būt divi apakšmezgli (“bērnu” mezgli), un vērtības tiek sakārtotas, pārbaudot vai jaunais elements ir lielāks vai mazāks par pamatelementu, ja mazāks tad tas veido kreiso atzaru, ja lielāks tad labo.



Uzdevums

Izveidot bināro koku.

Izveidot klasi *BNode* ar īpašībām:

- *number*- int tipa;
- *rightNode*– *BNode tipa*;
- *leftNode* - *BNode tipa.*

Konstruktors *public BNode(int Number)*, kurš inicializē *Number* īpašību.

Izveidot klasi *BTree* ar īpašībām:

- *rootNode* – *BNode* tipa*;*

Algoritmi un struktūras

BTree funkcijas:

- public Add (int n)- kura pievieno kokam jaunu mezglu;
- private Add(int n, Node ParentNode)- pievieno mezglam apakšmezglu;
- public Print()- izvada uz ekrānu koku pārskatāmā veidā;
- private Print(Node N, string seperator)- izvada konkrēto mezglu; kā atdalītāju izmantot padoto string tipa vērtību;
- public Remove(int number)- izdzēš mezglu, kura vērtība sakrīt ar padoto. Lai dzēšot saglabātos binārā meklēšanas koka īpašības, ņemt vērā šādus gadījumus:
    - ja dzēšamajam mezglam nav labā apakšmezgla, aizvietot dzēšamo mezglu ar tā kreiso apakšzaru;
    - ja dzēšamā mezgla labajam apakšmezglam nav kreisā mezgla, tad aizvietot dzēšamo mezglu ar tā labo mezglu;
    - ja dzēšamā mezgla labajam apakšmezglam ir kreisais mezgls, tad dzēšamo mezglu aizstāt ar tā labā mezgla pēdējo kreiso mezglu (jo tas satur labā mezgla apakšmezglu mazāko vērtību).

- public BNode FindNode(int n)- atgriež mezglu, kam ir norādītā vērtība
- public Walk(BNode nod)- rekursīvi apstaigā visus elementus, nosakot mezglu skaitu kokā.

Kokam jāsatur vienu saknes mezglu RootNode un int tipa mainīgo Skaits, kurā glabājas kokā esošo mezglu skaits.