Olympiad in Informatics

Trees

Instructor: Ha Viet Uyen Synh, Ph.D.

Build a binary tree

- Create(T, str, L): create a binary tree T from a string str follow a type L (LNR, NLR, LRN, NRL, RNL, RLN)
- LNR(T): in-order traversal on a binary tree T
- NLR(T): pre-order traversal on a binary tree T
- LRN(T): post-order traversal on a binary tree T

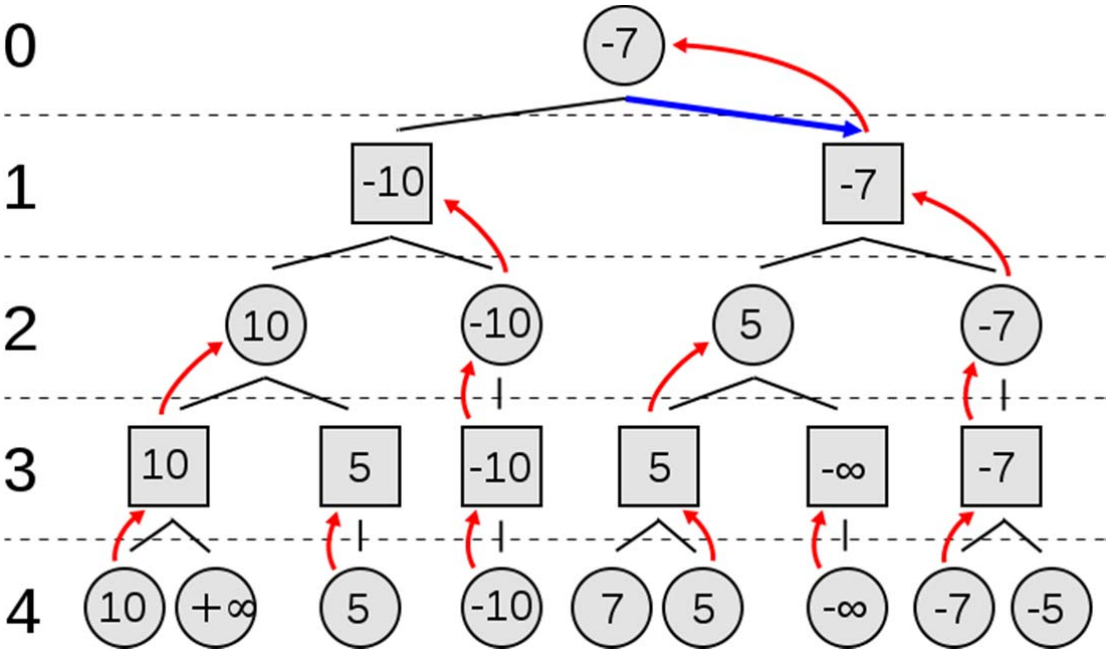
Build a binary search tree

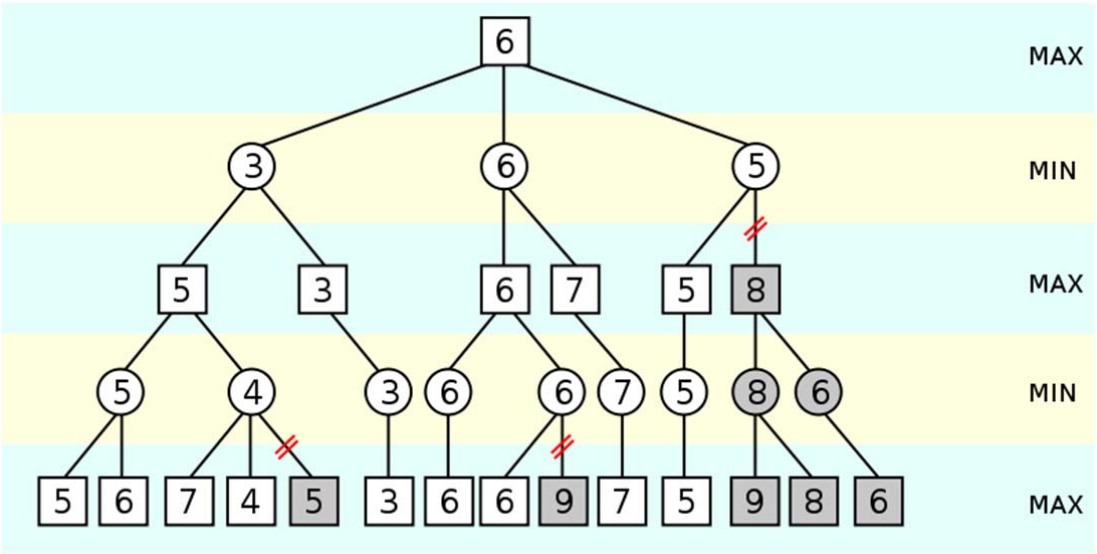
- Create(T, str)
- LNR(T)
- Insert(T, I): inseret a node I into a tree T
- Delete(T, I): remove a node I off a tree T
- Search(T, I): find a node I on a tree T

Build a balanced binary tree

- Create(T, str)
- LNR(T)
- Insert(T, I): inseret a node I into a tree T
- Delete(T, I): remove a node I off a tree T
- Search(T, I): find a node I on a tree T

1. Write a program that calculates the number of leaf nodes on a binary tree T.
2. Write a program that finds the depth of a binary tree.
3. Write a program that lists nodes on a binary tree follow each level.
4. *Write a function that describes the Min-Max algorithm.
5. *Write a function that describes the Alpha-Beta algorithm.
6. Write a Reversi / Caro / Life game (human vs computer)


0
1
2
3
4
-7
-10
-7
10
-10
5
-7
10
5
-10
5
-∞
-7
10
+∞
5
-10
7
5
-∞
-7
-5


MAX
MIN
MAX
MIN
MAX
6
3
6
5
5
3
6
7
5
8
5
4
3
6
6
7
5
8
6
5
6
7
4
5
3
6
6
9
7
5
9
8
6