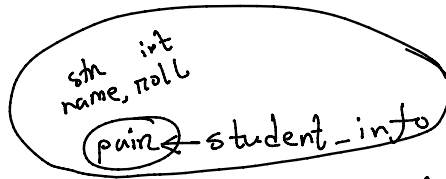


STL

vector ✓
queue ✓
stack ✓

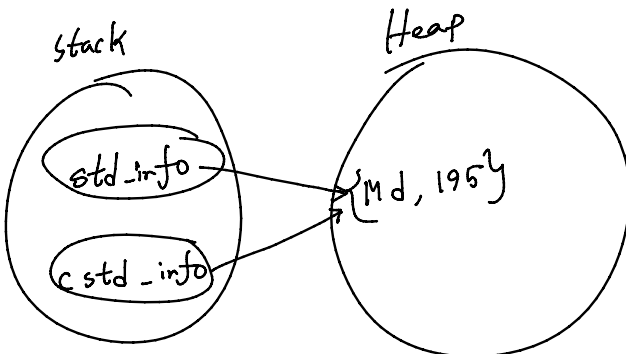
pair



LIST of student information

vector<pair<str, int>> students_info
↑
type

→ M.....
→ Zahangir
 ①s
 ①x



Sudoku →

Frontend
Backend
OOP (Design Pattern)
Pruning (Algo)
Cloud &
DB
Socket programming
Unit testing

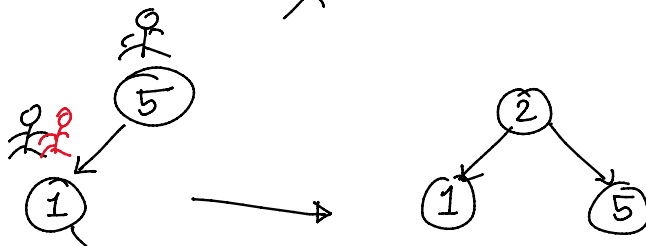
set

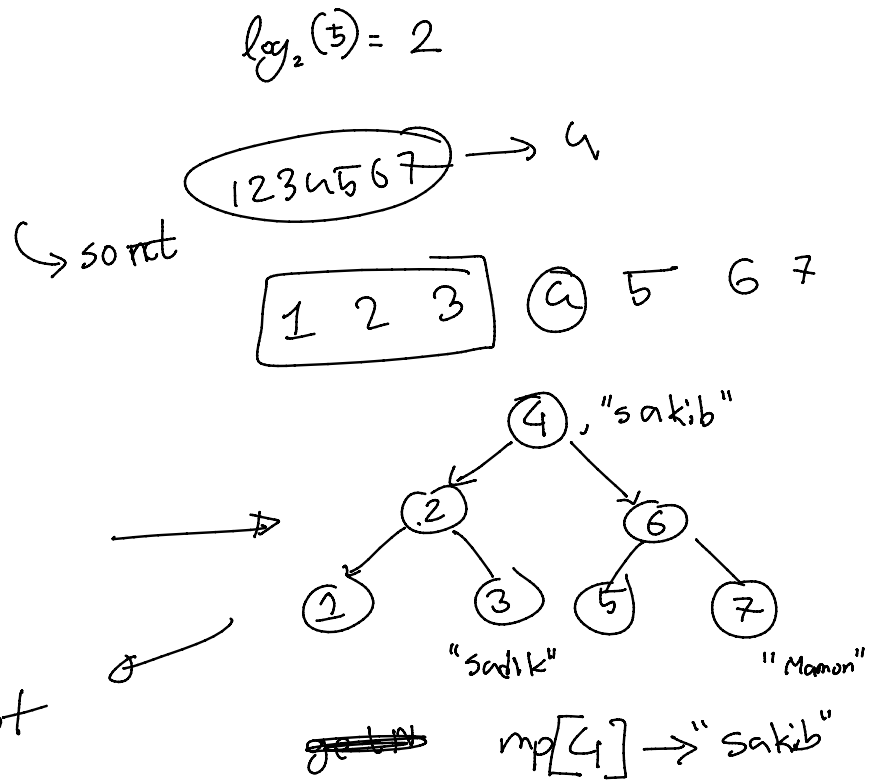
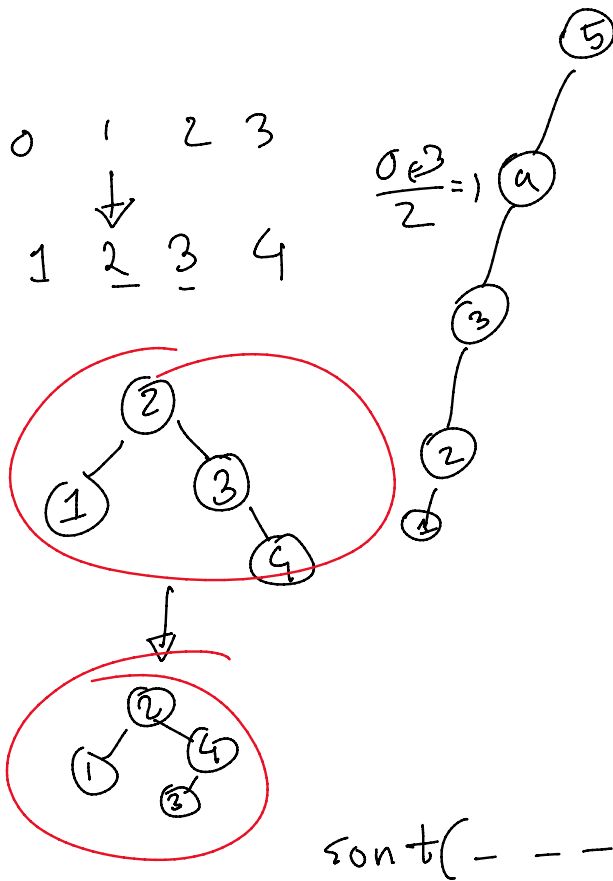
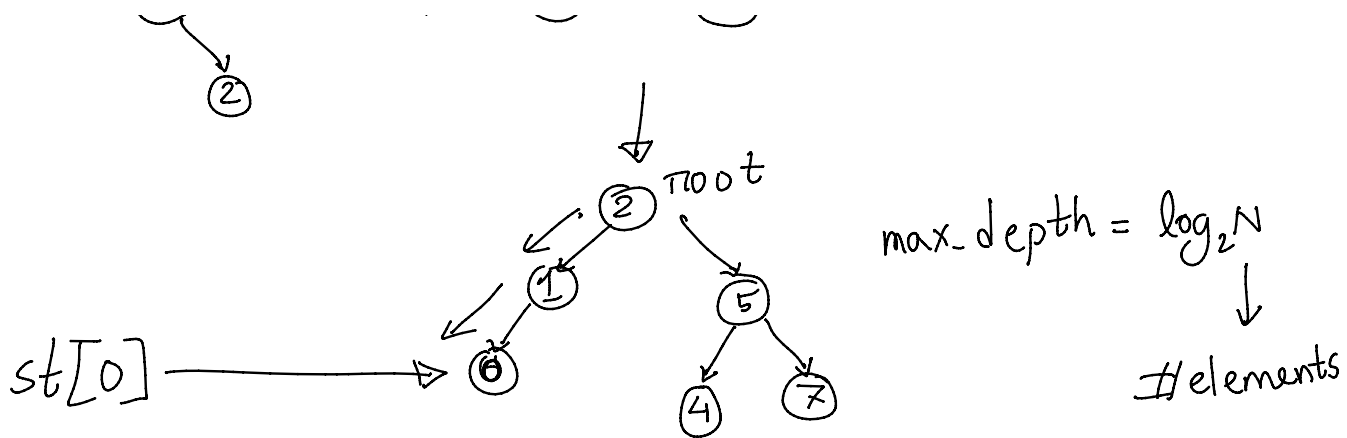
Red-black Tree

→ binary search tree (BST)

→ Self-balanced!

↓ ↓ ↓ ↓ ↓ ↓ ↓
5 1 2 0 7 2 4
 X





TreeNode* constructBST(vector<int> &elems, int L, int R)

```

{
    if (L > R) return null;
    int midIndex = (L + R) / 2;
    int midElem = elems[midIndex];
    TreeNode* curr = new TreeNode(midElem);
    curr->left = constructBST(elems, L, midIndex - 1);
    curr->right = constructBST(elems, midIndex + 1, R);
    return curr;
}

```

minimum
minimal

1 2 3 5 3 2 **6** 1

