Valid BST

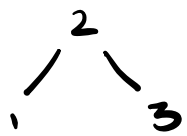
Asteroid Destruction

Winner Stone

Given root of a binary tree, check if it is a valid BST.

For every node x, all nodes in LST $< x$ and all nodes in RST $> x$.

Both LST and RST should be valid BST.



True

False

$1 \le N \le 10^5$

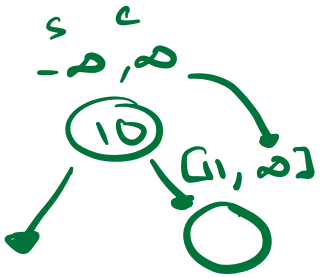
$0 \le val \le 2^{32} - 1$

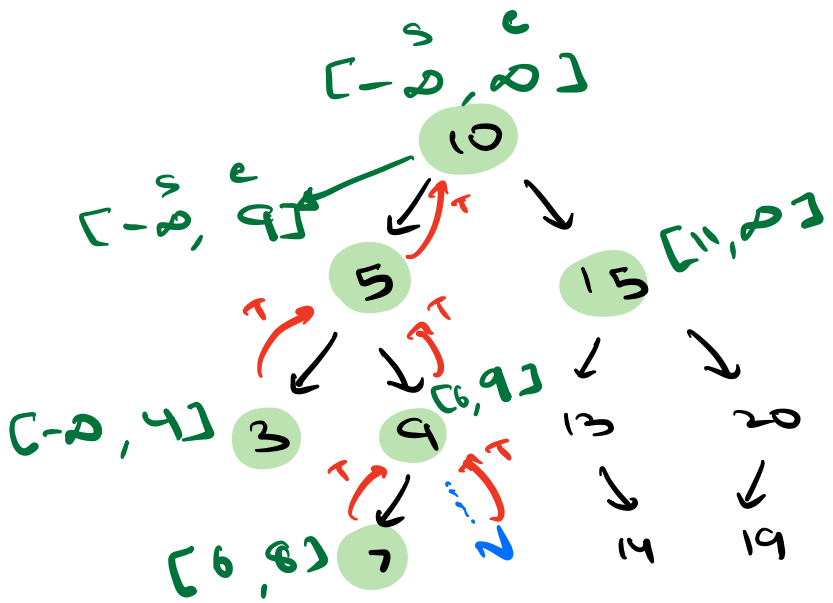
① Inorder traversal in BST → sorted

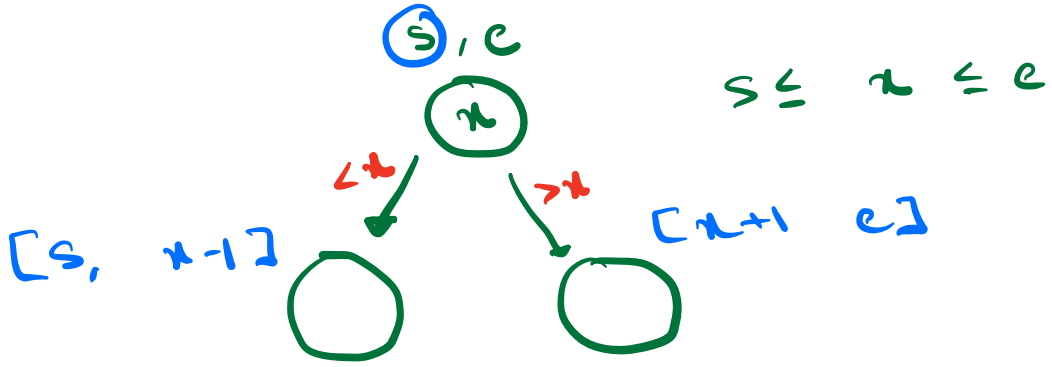
prev.val < A.val

② Preorder traversal

NLR







INT_MIN INT_MAX

A , $-\infty$, ∞

```
bool isValidBST (Node root, int s, int e){
    if (root == NULL)
        return true

    if (root->val >= s && root->val <= e){
        return
        (isValidBST (root->left,  s , root->val - 1)
          &&
          isValidBST (root->right, root->val+1, e))
    }
    return false
```

isValidBST (A, INT_MIN, INT_MAX)

TC: O(N)

SC: O(H) Fn call stack

↓

$\log N \to N$

Given a spaceship with energy A and N asteroids (i^th asteroid has energy B[i]).
Spaceship can collide with the asteroids in any order.

- If spaceship energy ≥ asteroid energy
  asteroid → destroyed
  spaceship gains asteroid's energy

- Otherwise spaceship is destroyed

Return 1 if all the asteroids are destroyed.

A = 10
B = [14, 3, 2]

ans = 1

10 → 2 → 12 → 3 → 15 → 14 → 29

10 → 3 → 13 → 2 → 15 → 14 → 29

Idea: Attack asteroids from smaller to bigger energy

A = 4
B = [2, 9, 2] —sort→ [2, 2, 9]

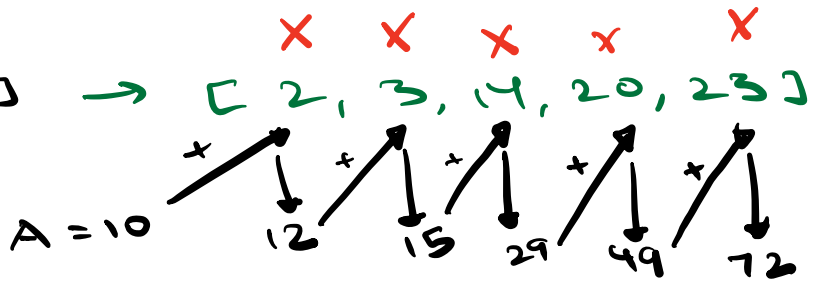
Can't attack 9

A = 4 → 6 → 8 → Return 0

A = 10

B = [14, 3, 2, 20, 23] → [2, 3, 14, 20, 23]

ans → 1




```
int destroyed(int A, int B[], int n){
    sort(B)
    long long int x = A
    for(i=0 ; i<n ; i++){
        if(x >= B[i]){
            x += B[i]   // A = A + B[i]
        }
        else
            return 0
    }
    return 1
}
```

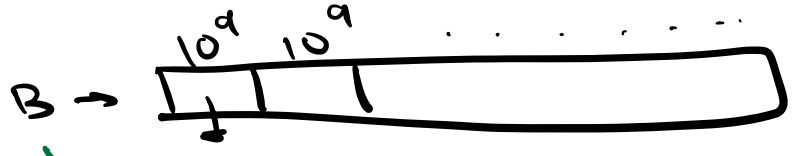
TC: O(N log N)

SC: O(N)

$1 \le A \le 10^9$

$1 \le N \le 10^5$

$1 \le B[i] \le 10^9$



sum = $10^9 \times 10^5 = 10^{14}$

Given an array A which represents weight of stones. On each turn, choose the heaviest 2 stones and smash them together.

Suppose stones have weight $x <= y$,

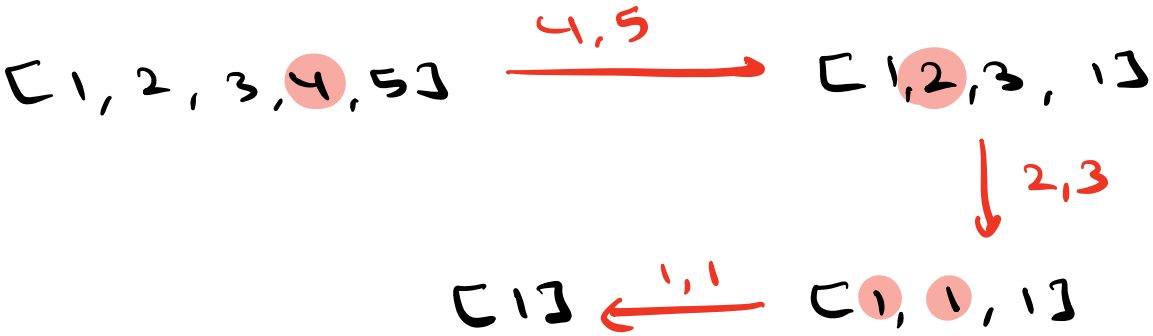
- If $x == y$, stones are destroyed
- If $x != y$, x is destroyed, stone of weight y has new weight $y - x$

At the end, there is at most 1 stone left. Return weight of the stone.

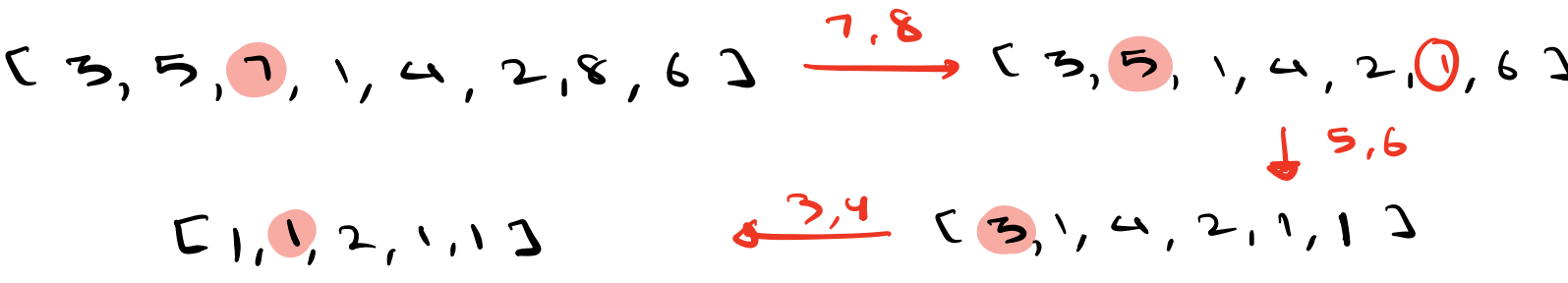
$1 \le N \le 10^4$

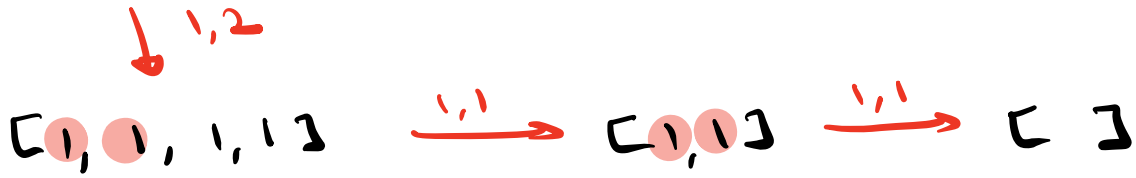
$1 \le A[i] \le 10^6$

A = [1, 2, 3, 4, 5] ans 1



A = [3, 5, 7, 1, 4, 2, 8, 6] ans 0





① Use a Max Heap

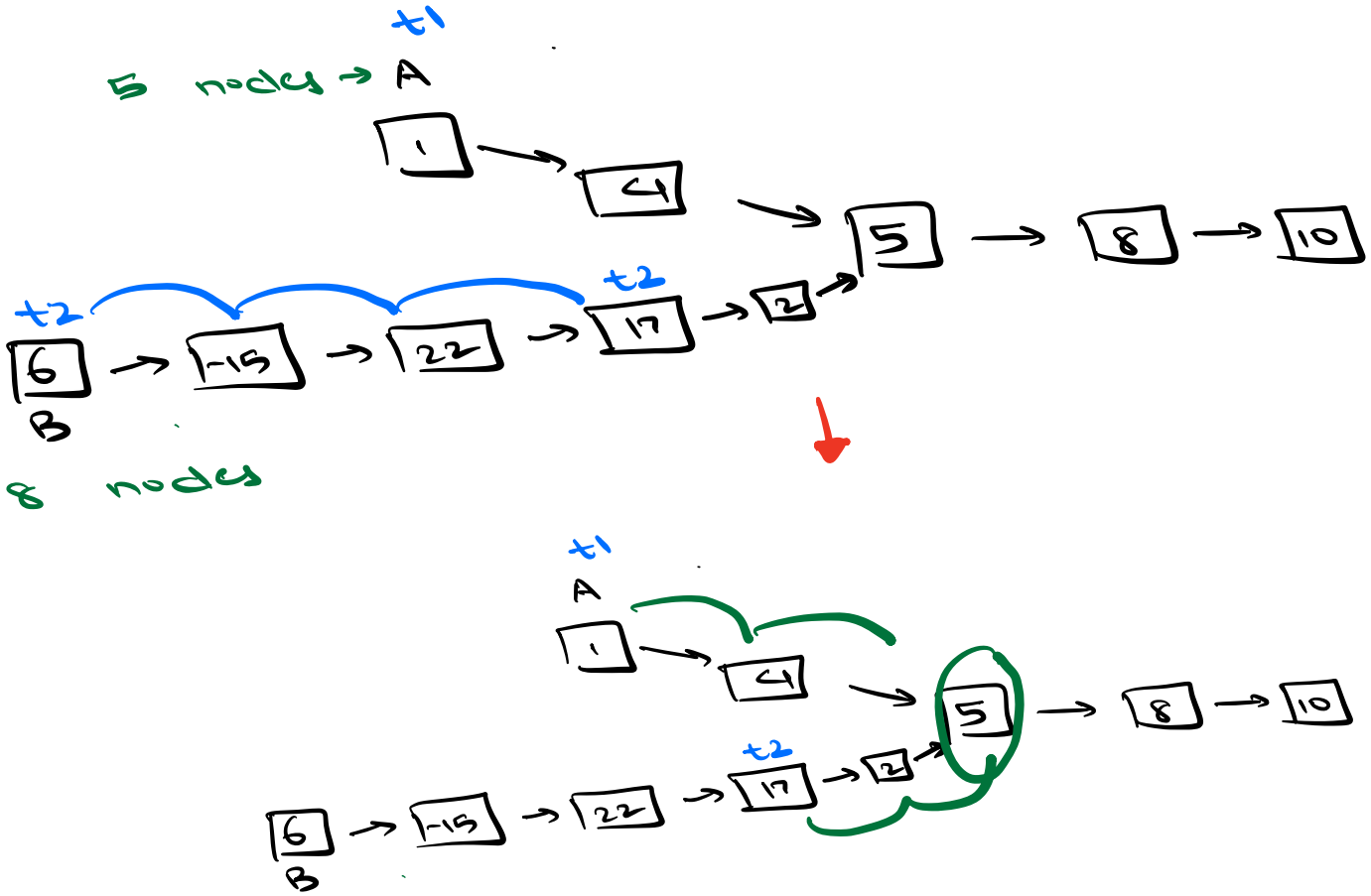
```
int lastStone(int A[]){

    Maxheap<int> mh
    for(i=0 ; i<n ; i++){
        mh.insert(A[i])
    }

    while(!mh.empty()){
        if(mh.size()==1)
            return mh.getMax()
        int y = mh.extractMax()
        int x = mh.extractMax()
        if(y!=x)
            mh.insert(y-x)
    }

    return 0
}
```

A → [5]

TC : O(N log N)

SC : O(N)

Doubts (Intersection of 2 LL)



1. Find size of LL1 and LL2
   x y

2. If y > x
   temp2 = LL2.head
   Make y - x jumps

3. temp1 = LL1.head
   temp2
   while (temp1 != temp2) {
   temp1 = temp1.next
   temp2 = temp2.next
   }
   return temp1