# 题目

给定一棵二叉搜索树，请找出其中第k大的节点。

**示例 1:**

输入: root = [3,1,4,null,2], k = 1

3

/ \

1 4

\

  2

输出: 4

**示例 2:**

输入: root = [5,3,6,2,4,null,null,1], k = 3

5

/ \

3 6

/ \

2 4

/

1

输出: 4

**限制：**

1 ≤ k ≤ 二叉搜索树元素个数

# 分析

## 方法一：优先队列/堆

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

priority\_queue<int> pri\_que;

void dfs(TreeNode\* root)

{

pri\_que.push(root->val);

if(root->left) dfs(root->left);

if(root->right) dfs(root->right);

}

int kthLargest(TreeNode\* root, int k) {

dfs(root);

while(--k)//不能使用k--(eg:[3,1,4,null,2])

{

pri\_que.pop();

}

return pri\_que.top();

}

};