

二叉搜索树的第 k 大节点

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

解题思路:

考虑到二叉搜索树的特性 左<根<右，按照中序遍历，实现从小到大排序

选取第 k 大数，即倒数第 k 个数即为最终结果

/**

```

* 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:
    vector<int> res;
    void dfs(TreeNode* root)//中序遍历 根据二叉树特性（左<根<右）从小到大排序
    {
        if(root==nullptr) return;
        dfs(root->left);
        res.push_back(root->val);
        dfs(root->right);
    }
    int kthLargest(TreeNode* root, int k) {
        dfs(root);
        return res[res.size()-k];
    }
};

```