

给定一个不含重复元素的整数数组。一个以此数组构建的最大二叉树定义如下：

1. 二叉树的根是数组中的最大元素。
2. 左子树是通过数组中最大值左边部分构造出的最大二叉树。
3. 右子树是通过数组中最大值右边部分构造出的最大二叉树。

通过给定的数组构建最大二叉树，并且输出这个树的根节点。

Example 1:

输入：[3,2,1,6,0,5]

输入：返回下面这棵树的根节点：



注意:

1. 给定的数组的大小在 [1, 1000] 之间。

思路，遍历一遍数组，找到其最大数的位置，然后动态申请需要的内存大小，用copy函数将数组分为左右两半传给函数，函数返回根节点。

```
/**
 * 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:
    TreeNode* dfs(vector<int> nums) {
```

```

    if(nums.size()==0) {
        return NULL;
    }
    int max=0;
    for(int i=1;i<nums.size();i++) {
        if(nums[i]>nums[max]) {
            max=i;
        }
    }
    TreeNode* root=new TreeNode(nums[max]);
    vector<int> left(max);
    vector<int> right(nums.size()-max-1);
    copy(nums.begin(),nums.begin()+max,left.begin());
    copy(nums.begin()+max+1,nums.end(),right.begin());
    root->left=dfs(left);
    root->right=dfs(right);
    return root;
}

TreeNode* constructMaximumBinaryTree(vector<int>& nums) {
    return dfs(nums);
}

};

```