

```

1  /*
2  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：
3
4      二叉树的根是数组中的最大元素。
5      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。
6      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。
7
8  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。
9
10
11
12  示例：
13
14  输入：[3,2,1,6,0,5]
15  输出：返回下面这棵树的根节点：
16
17      6
18     / \
19    3   5
20     \ /
21     2 0
22      \
23       1
24
25
26
27  提示：
28
29      给定的数组的大小在 [1, 1000] 之间。
30
31  来源：力扣（LeetCode）
32  链接：https://leetcode-cn.com/problems/maximum-binary-tree
33  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
34  */

```

分析:

- 递归解法:和快速排序的流程相似,找出最值点然后分块处理.

方法一: C++_递归法

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     struct TreeNode *left;
6   *     struct TreeNode *right;
7   * };
8   */
9  struct TreeNode* __maxTree( int*   nums   ,
10                             int    left   ,
11                             int    right  )
12

```

```

13 {
14
15     struct TreeNode* ret_val = NULL ;
16     int ind_max = left ;
17     int i = 0 ;
18
19     do
20     {
21         if(left>right)
22         {
23             break;
24         }
25
26         ret_val = (struct TreeNode*)malloc(sizeof(struct TreeNode));
27         if(left == right)
28         {
29             ret_val->val = nums[left];
30             ret_val->left = NULL;
31             ret_val->right = NULL;
32             break;
33         }
34
35         for(i=left+1;i<=right;i++)
36         {
37             if(nums[i] > nums[ind_max])
38             {
39                 ind_max = i;
40             }
41         }
42         ret_val->val = nums[ind_max];
43         ret_val->left = __maxTree(nums, left, ind_max-1);
44         ret_val->right = __maxTree(nums, ind_max+1, right);
45         break;
46     }while(0);
47
48     return ret_val;
49 }
50
51
52
53 struct TreeNode* constructMaximumBinaryTree( int* nums ,
54                                              int numSize
55                                              )
56 {
57     return __maxTree(nums,0,numSize-1);
58 }
59
60
61 /*
62 执行结果:
63 通过
64 显示详情
65 执行用时 :72 ms, 在所有 C 提交中击败了57.85% 的用户
66 内存消耗 :28.3 MB, 在所有 C 提交中击败了100.00%的用户
67 */

```

