

```

1  /*
2  给定一个二叉树，在树的最后一行找到最左边的值。
3  示例 1:
4  输入:
5      2
6     /\
7    1  3
8  输出:
9  1
10 示例 2:
11 输入:
12      1
13     /\
14    2  3
15   /\ /\
16  4  5 6
17   /\
18  7
19 输出:
20 7
21 注意：您可以假设树（即给定的根节点）不为 NULL。
22 */

```

分析:

常规题目:在BFS过程中,总是保存每层第一个节点的值即可.

方法一:C++\_BFS

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9   */
10 class Solution
11 {
12     public:
13     int findBottomLeftValue(TreeNode* root)
14     {
15
16         queue<TreeNode*>    qtn                ;
17         int                 ret_val = 0        ;
18         TreeNode*          temp              ;
19         int                 num_1  = 0         ;
20         int                 num_2  = 1         ;
21         int                 i       = 0         ;
22         int                 j       = 0         ;
23
24         qtn.push(root);

```

```

25         while(!qtn.empty())
26         {
27             num_1 = num_2;
28             num_2 = 0;
29             ret_val = qtn.front()->val; /*行首的指针节点的值*/
30             for(i=0;i<num_1;i++)
31             {
32                 temp = qtn.front();
33                 qtn.pop();
34                 if(temp->left)
35                 {
36                     qtn.push(temp->left);
37                     num_2++;
38                 }
39                 if(temp->right)
40                 {
41                     qtn.push(temp->right);
42                     num_2++;
43                 }
44             }
45         }
46         return ret_val;
47     }
48 };
49
50 /*
51 执行结果:
52 通过
53 显示详情
54 执行用时 :16 ms, 在所有 C++ 提交中击败了88.68% 的用户
55 内存消耗 :20.8 MB, 在所有 C++ 提交中击败了66.24%的用户
56 */

```