

```

1  /*
2  给定一个保存员工信息的数据结构，它包含了员工唯一的id，重要度 和 直系下属的id。
3
4  比如，员工1是员工2的领导，员工2是员工3的领导。他们相应的重要度为15，10，5。那么员工1的数据结构是[1, 15, [2]]，员工2的数据结构是[2, 10, [3]]，员工3的数据结构是[3, 5, []]。注意虽然员工3也是员工1的一个下属，但是由于并不是直系下属，因此没有体现在员工1的数据结构中。
5
6  现在输入一个公司的所有员工信息，以及单个员工id，返回这个员工和他所有下属的重要度之和。
7
8  示例 1:
9
10 输入: [[1, 5, [2, 3]], [2, 3, []], [3, 3, []]], 1
11 输出: 11
12 解释:
13 员工1自身的重要度是5，他有两个直系下属2和3，而且2和3的重要度均为3。因此员工1的总重要度是
14 5 + 3 + 3 = 11。
15
16 注意:
17     一个员工最多有一个直系领导，但是可以有多个直系下属
18     员工数量不超过2000。
19
20 通过次数9,385
21 提交次数16,439
22
23 来源：力扣（LeetCode）
24 链接：https://leetcode-cn.com/problems/employee-importance
25 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。
26 */

```

分析：

- 方法一：依次递归循环遍历数组id直到下属数组遍历完成或下属数组为空为止。

方法一:C++_循环递归遍历

```

1  /*
2  // Employee info
3  class Employee {
4  public:
5      // It's the unique ID of each node.
6      // unique id of this employee
7      int id;
8      // the importance value of this employee
9      int importance;

```

```

10     // the id of direct subordinates
11     vector<int> subordinates;
12 };
13 */
14 class solution
15 {
16
17     private:
18
19         int __getImportance(vector<Employee*>& employees, int id)
20         {
21             int size1 = employees.size();
22             int ret_val = 0;
23             for(int i = 0 ; i < size1 ; i++)
24             {
25                 if(id == employees[i]->id)
26                 {
27                     ret_val += employees[i]->importance;
28                     int size2 = employees[i]->subordinates.size();
29                     for(int j = 0 ; j < size2 ; j++ )
30                     {
31                         ret_val += __getImportance(employees,employees[i]-
32 >subordinates[j]);
33                     }
34                     break;
35                 }
36             }
37             return ret_val;
38         }
39
40     public:
41         int getImportance(vector<Employee*> employees, int id)
42         {
43             return __getImportance(employees,id);
44         }
45 };
46
47
48
49 /*
50 执行结果:
51 通过
52 显示详情
53 执行用时 :56 ms, 在所有 C++ 提交中击败了25.43% 的用户
54 内存消耗 :14.1 MB, 在所有 C++ 提交中击败了96.56%的用户
55 */

```

