

```

1  /*
2  集合的表示及查找
3
4  并查集
5      把集合并起来，然后查某个元素属于什么集合
6
7      例子：
8          有 10 台电脑，然后某些电脑之间直接相连
9          问：某两台不是直接相连的电脑是否存在连接？
10     解决思路：
11         (1) 将10台电脑看成10个集合{1}，{2}，{3}，...，{10}
12         (2) 如果两台电脑存在连接，就将两台电脑属于的集合合并在一起
13         (3) 查询两台电脑是否连接，就是判别他们是否属于同一集合
14
15     可以用树结构表示集合，树的每个结点表示一个集合元素
16         (1) 使用链表存储：树使用双亲表示法，孩子指向双亲
17         (2) 使用数组存储，采用结构体，存储结点的值和双亲下标
18 */
19
20 #include<iostream>
21 #include<cstring>
22 #define MaxSize 10
23 using namespace std;
24
25 typedef int ElementType;
26 typedef struct{
27     ElementType Data; // 存值
28     int parent;       // 指向父结点在数组中的下标
29 }SetType;
30
31 // 查找某个元素所在的集合，该集合用树的根结点表示
32 int Find(SetType s[], ElementType x)
33 {
34     int i;
35     // 找到数组中该值对应的下标
36     for(i=0; i<MaxSize&& s[i].Data!=x; i++);
37     if(i >= MaxSize) // 如果没有找到，返回 -1
38         return -1;
39     // 找到该结点的根结点
40     for(; s[i].parent>=0; i=s[i].parent);
41     return i; // 返回根结点在数组 s 中的下标
42 }
43
44 // 并
45 void Union(SetType s[], ElementType x1, ElementType x2)
46 {
47     int root1 = Find(s, x1); // 找到 x1 的根结点下标
48     int root2 = Find(s, x2); // 找到 x2 的根结点下标
49     // 如果根结点的下标不同，说明不是一个集合
50     if(root1 != root2)
51     {
52         s[root1].parent = root2; // 把 x1 挂到 x2 的集合
53     }
54 }
55
56 int main()
57 {
58     SetType s[MaxSize];
59     // 初始化数组，父结点全部指向 -1
60     for(int i=0; i<MaxSize; i++)
61     {
62         s[i].Data = i + 1;
63         s[i].parent = -1;
64     }
65     // 如果父结点指向 -1，该节点所属集合是其自身的下标
66     cout<<Find(s, 5)<<endl; // 4
67     Union(s, 3, 5);
68     cout<<Find(s, 4)<<endl; // 3
69     cout<<Find(s, 3)<<endl; // 4
70     Union(s, 1, 3); // {1, 3, 5}
71     Union(s, 2, 4); // {2, 4}
72     Union(s, 8, 6); // {8, 6}
73     cout<<Find(s, 6)<<endl; // 5

```

```
74     cout<<Find(s, 8)<<endl; // 5
75     return 0;
76 }
77
```