

连通块中点的数量

给定一个包含n个点（编号为1~n）的无向图，初始时图中没有边。

现在要进行m个操作，操作共有三种：

- “C a b”，在点a和点b之间连一条边，a和b可能相等；
- “Q1 a b”，询问点a和点b是否在同一个连通块中，a和b可能相等；
- “Q2 a”，询问点a所在连通块中点的数量；

输入格式

第一行输入整数n和m。

接下来m行，每行包含一个操作指令，指令为“C a b”，“Q1 a b”或“Q2 a”中的一种。

输出格式

对于每个询问指令“Q1 a b”，如果a和b在同一个连通块中，则输出“Yes”，否则输出“No”。

对于每个询问指令“Q2 a”，输出一个整数表示点a所在连通块中点的数量

每个结果占一行。

数据范围

$$1 \leq n, m \leq 10^5$$

输入样例：

```
5 5
C 1 2
Q1 1 2
Q2 1
C 2 5
Q2 5
```

输出样例：

```
Yes
2
3
```

y总代码：

```

12     if (p[x] != x) p[x] = find(p[x]);
13     return p[x];
14 }
15
16 int main()
17 {
18     scanf("%d%d", &n, &m);
19
20     for (int i = 1; i <= n; i++)
21     {
22         p[i] = i;
23         size[i] = 1;
24     }
25
26     while (m--)
27     {
28         char op[5];
29         int a, b;
30         scanf("%s", op);
31
32         if (op[0] == 'c')
33         {
34             scanf("%d%d", &a, &b);
35             if (find(a) == find(b)) continue;
36             size[find(b)] += size[find(a)];
37             p[find(a)] = find(b);
38         }
39         else if (op[1] == 'i')
40         {
41             scanf("%d%d", &a, &b);
42             if (find(a) == find(b)) puts("Yes");
43             else puts("No");
44         }
45         else
46         {
47             scanf("%d", &a);
48             printf("%d\n", size[find(a)]);
49         }
50     }
51
52     return 0;

```

```

#include <iostream>

using namespace std;

const int N = 100010;

int n, m;
int p[N], cnt[N];

int find(int x)
{
    if (p[x] != x) p[x] = find(p[x]);
    return p[x];
}

int main()
{
    cin >> n >> m;

    for (int i = 1; i <= n; i++)
    {
        p[i] = i;
        cnt[i] = 1;
    }

    while (m--)
    {
        string op;
        int a, b;
        cin >> op;

        if (op == "c")
        {

```

```

        cin >> a >> b;
        a = find(a), b = find(b);
        if (a != b)
        {
            p[a] = b;
            cnt[b] += cnt[a];
        }
    }
    else if (op == "Q1")
    {
        cin >> a >> b;
        if (find(a) == find(b)) puts("Yes");
        else puts("No");
    }
    else
    {
        cin >> a;
        cout << cnt[find(a)] << endl;
    }
}

return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/45295/>

来源: Acwing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。