

合并集合

一共有 n 个数，编号是 $1 \sim n$ ，最开始每个数各自在一个集合中。

现在要进行 m 个操作，操作共有两种：

1. “M a b”，将编号为 a 和 b 的两个数所在的集合合并，如果两个数已经在同一个集合中，则忽略这个操作；
2. “Q a b”，询问编号为 a 和 b 的两个数是否在同一个集合中；

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“M a b”或“Q a b”中的一种。

输出格式

对于每个询问指令“Q a b”，都要输出一个结果，如果 a 和 b 在同一集合内，则输出“Yes”，否则输出“No”。

每个结果占一行。

数据范围

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

输入样例：

```
4 5
M 1 2
M 3 4
Q 1 2
Q 1 3
Q 3 4
```

输出样例：

```
Yes
No
Yes
```

并查集：

1. 将两个集合合并
2. 询问两个元素是否在一个集合当中

基本原理：每个集合用一棵树来表示。树根的编号就是整个集合的编号。每个节点存储它的父节点， $p[x]$ 表示 x 的父节点

问题1：如何判断树根：if ($p[x] == x$)

问题2：如何求 x 的集合编号：while ($p[x] != x$) $x = p[x]$;

问题3：如何合并两个集合： px 是 x 的集合编号， py 是 y 的集合编号。 $p[x] = y$

y总代码：

```

#include <iostream>

using namespace std;

const int N = 100010;

int p[N];

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

int main()
{
    int n, m;
    scanf("%d%d", &n, &m);
    for (int i = 1; i <= n; i ++ ) p[i] = i;

    while (m -- )
    {
        char op[2];
        int a, b;
        scanf("%s%d%d", op, &a, &b);
        if (*op == 'M') p[find(a)] = find(b);
        else
        {
            if (find(a) == find(b)) puts("Yes");
            else puts("No");
        }
    }

    return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/45287/>

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。