

一：合并集合

无任何优化

```

#include <bits/stdc++.h>

using namespace std;

int fa[100010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
    }
}

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

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx != fy){
        fa[fx] = fy;
    }
}

int main(){
    int n, m;
    cin >> n >> m;
    init(n);
    while(m -- ){
        char op;
        int a, b;
        cin >> op >> a >> b;
        if(op == 'M'){
            merge(a, b);
        } else {
            if(find(a) == find(b)){
                puts("Yes");
            } else {
                puts("No");
            }
        }
    }

    return 0;
}

```

路径压缩

```
#include <bits/stdc++.h>

using namespace std;

int fa[100010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
    }
}

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

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx != fy){
        fa[fx] = fy;
    }
}

int main(){
    int n, m;
    cin >> n >> m;
    init(n);
    while(m -- ){
        char op;
        int a, b;
        cin >> op >> a >> b;
        if(op == 'M'){
            merge(a, b);
        } else {
            if(find(a) == find(b)){
                puts("Yes");
            } else {
                puts("No");
            }
        }
    }

    return 0;
}
```

按秩合并

```

#include <bits/stdc++.h>

using namespace std;

int fa[100010];
int sz[100010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
        sz[i] = 1;
    }
}

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

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx == fy){
        return ;
    }
    if(sz[fx] > sz[fy]){
        swap(fx, fy);
    }
    fa[fx] = fy;
    sz[fy] += sz[fx];
}

int main(){
    int n, m;
    cin >> n >> m;
    init(n);
    while(m -- ){
        char op;
        int a, b;
        cin >> op >> a >> b;
        if(op == 'M'){
            merge(a, b);
        } else {
            if(find(a) == find(b)){
                puts("Yes");
            } else {
                puts("No");
            }
        }
    }
}

```

```
        }  
    }  
    return 0;  
}
```

路径压缩 + 按秩合并

```

#include <bits/stdc++.h>

using namespace std;

int fa[100010];
int sz[100010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
        sz[i] = 1;
    }
}

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

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx == fy){
        return ;
    }
    if(sz[fx] > sz[fy]){
        swap(fx, fy);
    }
    fa[fx] = fy;
    sz[fy] += sz[fx];
}

int main(){
    int n, m;
    cin >> n >> m;
    init(n);
    while(m -- ){
        char op;
        int a, b;
        cin >> op >> a >> b;
        if(op == 'M'){
            merge(a, b);
        } else {
            if(find(a) == find(b)){
                puts("Yes");
            } else {
                puts("No");
            }
        }
    }
}

```

```
    }  
    }  
    return 0;  
}
```

二：集合中点的数量


```

#include <bits/stdc++.h>

using namespace std;

int fa[100010];
int sz[100010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
        sz[i] = 1;
    }
}

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

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx == fy){
        return ;
    }
    if(sz[fx] > sz[fy]){
        swap(fx, fy);
    }
    fa[fx] = fy;
    sz[fy] += sz[fx];
}

int main(){
    int n, m;
    cin >> n >> m;
    init(n);
    while(m -- ){
        string op;
        int a, b;
        cin >> op >> a;
        if(op == "C"){
            cin >> b;
            merge(a ,b);
        } else if(op == "Q1"){
            cin >> b;
            if(find(a) == find(b)){
                puts("Yes");
            } else {

```

```
        puts("No");
    }
} else {
    cout << sz[find(a)] << '\n';
}
}
return 0;
}
```

三：亲戚

```

#include <bits/stdc++.h>

using namespace std;

int fa[20010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
    }
}

int find(int x){
    return (fa[x] == x) ? x : (fa[x] = find(fa[x]));
}

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx != fy){
        fa[fx] = fy;
    }
}

int main(){
    int n, m;
    scanf("%d %d", &n, &m);
    init(n);
    while(m -- ){
        int x, y;
        scanf("%d %d", &x, &y);
        merge(x, y);
    }
    int q;
    scanf("%d", &q);
    while(q -- ){
        int x, y;
        scanf("%d %d", &x, &y);
        if(find(x) == find(y)){
            puts("Yes");
        } else {
            puts("No");
        }
    }

    return 0;
}

```

四：冗余关系

```

#include <bits/stdc++.h>

using namespace std;

int fa[20010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
    }
}

int find(int x){
    return (fa[x] == x) ? x : (fa[x] = find(fa[x]));
}

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx != fy){
        fa[fx] = fy;
    }
}

int main(){
    int n, m;
    scanf("%d %d", &n, &m);
    init(n);
    while(m -- ){
        int x, y;
        scanf("%d %d", &x, &y);
        merge(x, y);
    }
    int ans = 0;
    for(int i = 1; i <= n; i ++ ){
        if(fa[i] == i){
            ans ++;
        }
    }
    printf("%d", ans);
    return 0;
}

```

五：通讯系统

```

#include <bits/stdc++.h>

using namespace std;

int fa[10010];

void init(int n){
    for(int i = 1; i <= n; i ++ ){
        fa[i] = i;
    }
}

int find(int x){
    return (fa[x] == x) ? x : (fa[x] = find(fa[x]));
}

void merge(int x, int y){
    int fx = find(x);
    int fy = find(y);
    if(fx != fy){
        fa[fx] = fy;
    }
}

int main(){
    int n, m;
    while(cin >> n >> m, n || m){
        init(n);
        for(int i = 1; i <= m; i ++ ){
            int x, y;
            std::cin >> x >> y;
            merge(x, y);
        }
        if(m > n - 1){
            puts("No");
            continue;
        }
        int ans = 0;
        for(int i = 1; i <= n; i ++ ){
            if(fa[i] == i){
                ans ++;
            }
        }
        if(ans == 1){
            puts("Yes");
        } else {
            puts("No");
        }
    }
}

```

```
}    return 0;
```