*int* n, m;

    std::cin >> n >> m;

    std::vector<*int*> fa(n + 1);

    std::iota(fa.begin(), fa.end(), 0);

    std::function<*int*(*int*)> find = [&](*int* i) -> *int* {

*if*(fa[i] != i)

            fa[i] = find(fa[i]);

*return* fa[i];

    };

*for*(*int* i = 1; i <= m; i++)

    {

*char* control;

        std::cin >> control;

*int* x, y;

        std::cin >> x >> y;

*if*(control == 'M')

        {

            fa[find(x)] = find(y);

        }

*else*

        {

*if*(find(x) == find(y))

                std::cout << "Yes" << endl;

*else*

                std::cout << "No" << endl;

        }

    }

并查集的代码很短，关键在于路径压缩的核心代码find

std::function<*int*(*int*)> find = [&](*int* i) -> *int* {

*if*(fa[i] != i)

            fa[i] = find(fa[i]);

*return* fa[i];

    };