

E. CUTTING TREE

PROBLEM DESCRIPTION

題目給一棵樹，並要求支援兩種操作：刪除邊、查詢兩點是否連通。

SOLUTION TECHNIQUES

Disjoint Set, 離線處理

SOLUTION SKETCHES

離線處理。

我們可以把查詢倒過來做，先假設所有刪除操作都已經執行了，然後我們可以從最後一個操作開始處理，若遇到刪除我們就把時光倒流：把邊黏回去，這樣一來再遇到查詢操作時，我們手上的樹即是當時的狀態。

判斷兩點連通可使用 Disjoint Set。

TIME COMPLEXITY

每筆測資 $O(Q(\log Q + \log N))$ ， N 為樹上點的數量、 Q 為查詢數量。

SOLUTION PROGRAM FOR REFERENCE

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <algorithm>
#include <vector>

using namespace std;
typedef pair<int, int> pii;

const int N = 20005;
const int K = 5005;

pii que[K];
vector<int> ans;
int p[N];
int f[N];
char del[N];
char used[N];

int findF(int vi)
{
    return f[vi] == vi ? vi : (f[vi] = findF(f[vi]));
}

void uniF(int vi, int vj)
{
    f[findF(vj)] = findF(vi);
}
```

```

int main()
{
    int i, j, tt, v, q;
    char typ; int vi, vj;
    scanf("%d", &tt);
    for (int cc = 1; cc <= tt; cc++)
    {
        scanf("%d%d", &v, &q);
        ans.clear();
        for (i = 1; i <= v; i++) f[i] = i;
        fill(del+1, del+1+v, 0);
        fill(used+1, used+1+v, 0);
        for (i = 1; i <= v; i++) scanf("%d", &p[i]);
        for (i = 0; i < q; i++)
        {
            scanf(" %c", &typ);
            if (typ == 'Q')
                scanf("%d%d", &que[i].first, &que[i].second);
            else
            {
                scanf("%d", &que[i].first);
                if (used[que[i].first]) { que[i].first =
que[i].second = -1; continue; }
                que[i].second = -1;
                del[que[i].first] = 1;
                used[que[i].first] = 1;
            }
        }
        for (i = 1; i <= v; i++)

```

```

        if (!del[i] && p[i])
            uniF(i, p[i]);
    for (i = q - 1; i >= 0; i--)
    {
        if (que[i].first == -1) continue;
        if (que[i].second != -1)
            ans.push_back(findF(que[i].first) ==
findF(que[i].second));
        else
        {
            if (p[que[i].first])
                uniF(que[i].first, p[que[i].first]);
        }
    }
    printf("Case #%d:\n", cc);
    for (i = ans.size() - 1; i >= 0; i--)
        puts(ans[i] ? "YES" : "NO");
}
return 0;
}

```