Contents

vector 建樹找尋最近共同祖先

```cpp
//#include<bits/stdc++.h>
#include<iostream>
#include<vector>
#include<cstring>
using namespace std;
const int N = 10000;
vector<int> a[N];
int f[N], r[N];
void DFS(int u, int dep)
{
    r[u] = dep;
    for(vector<int>::iterator it =
        a[u].begin(); it != a[u].end(); it++)
        DFS(*it, dep + 1);
}
int main()
{

    int casenum, num, n, i, x, y;
    scanf("%d", &casenum);
    for(num=0; num<casenum; num++)
    {
        scanf("%d", &n);
        for(i=0; i<n; i++) a[i].clear();
        memset(f, 255, sizeof(f));
        for(i=0; i<n-1; i++)
        {
            scanf("%d %d", &x, &y);
            a[x-1].push_back(y-1);
            f[y-1] = x-1;
        }
        for(i=0; f[i]>=0; i++);
        DFS(i, 0);
        scanf("%d %d", &x, &y);
        x--; y--;
        while(x != y)
        {
            if(r[x] > r[y]) x = f[x];
            else y = f[y];
        }
        printf("%d\n", x+1);
    }
    return 0;
}
```

disjoinset 並查集路徑壓縮

```cpp
#include<iostream>
#include<vector>
#include<cstring>
const int maxn = 100000+5;
int n, m;
int set[maxn + maxn];
int set_find(int d)
{

    if(set[d] < 0)
        return d;
    return set[d] = set_find(set[d]);
}
int main(void)
{

    int loop;
    scanf("%d", &loop);
    while(loop--)
    {
        scanf("%d%d", &n, &m);
        memset(set, -1, sizeof(set));
        for(int i=0; i<m; i++)
        {
            int a, b;
            char s[5];
            scanf("%s%d%d", s, &a, &b);
            if(s[0] == 'A')
            {
                if(set_find(a) != set_find(b) &&
                    set_find(a) != set_find(b+n))
                    printf("%s\n", "Not sure
                        yet.");
                else if(set_find(a) ==
                    set_find(b))
                    printf("%s\n", "In the same
                        gang.");
                else
                    printf("%s\n", "In different
                        gangs.");
            }
            else
            {
                if(set_find(a) != set_find(b+n))
                {
                    set[set_find(a)] =
                        set_find(b+n);
                    set[set_find(b)] =
                        set_find(a+n);
                }
            }
        }
    }
    return 0;
}
```

vector 建樹找尋最近共同祖先

```cpp
//#include<bits/stdc++.h>
#include<iostream>
#include<vector>
#include<cstring>
using namespace std;
const int N = 10000;
```

```cpp
vector<int> a[N];
int f[N], r[N];
void DFS(int u, int dep)
{
    r[u] = dep;
    for(vector<int>::iterator it =
        a[u].begin(); it != a[u].end(); it++)
       DFS(*it, dep + 1);
}
int main()
{

    int casenum, num, n, i, x, y;
    scanf("%d", &casenum);
    for(num=0; num<casenum; num++)
    {
        scanf("%d", &n);
        for(i=0; i<n; i++) a[i].clear();
        memset(f, 255, sizeof(f));
        for(i=0; i<n-1; i++)
        {
            scanf("%d %d", &x, &y);
            a[x-1].push_back(y-1);
            f[y-1] = x-1;
        }
        for(i=0; f[i]>=0; i++);
        DFS(i, 0);
        scanf("%d %d", &x, &y);
        x--; y--;
        while(x != y)
        {
            if(r[x] > r[y]) x = f[x];
            else y = f[y];
        }
        printf("%d\n", x+1);
    }
    return 0;
}
```