

Contents

1	vector 建樹找尋最近共同祖先	2
2	disjoinset 並查集路徑壓縮	2
3	vector 建樹找尋最近共同祖先	2

vector 建樹找尋最近共同祖先

```

1  // #include <bits/stdc++.h>
2  #include <iostream>
3  #include <vector>
4  #include <cstring>
5  using namespace std;
6  const int N = 10000;
7  vector<int> a[N];
8  int f[N], r[N];
9  void DFS(int u, int dep)
10 {
11     r[u] = dep;
12     for(vector<int>::iterator it =
13         a[u].begin(); it != a[u].end(); it++)
14         DFS(*it, dep + 1);
15 }
16 int main()
17 {
18     int casenum, num, n, i, x, y;
19     scanf("%d", &casenum);
20     for(num=0; num<casenum; num++)
21     {
22         scanf("%d", &n);
23         for(i=0; i<n; i++) a[i].clear();
24         memset(f, 255, sizeof(f));
25         for(i=0; i<n-1; i++)
26         {
27             scanf("%d %d", &x, &y);
28             a[x-1].push_back(y-1);
29             f[y-1] = x-1;
30         }
31         for(i=0; f[i]>=0; i++);
32         DFS(i, 0);
33         scanf("%d %d", &x, &y);
34         x--; y--;
35         while(x != y)
36         {
37             if(r[x] > r[y]) x = f[x];
38             else y = f[y];
39         }
40         printf("%d\n", x+1);
41     }
42     return 0;
43 }

```

disjoinset 並查集路徑壓縮

```

1  #include <iostream>
2  #include <vector>
3  #include <cstring>
4  const int maxn = 100000+5;
5  int n, m;
6  int set[maxn + maxn];
7  int set_find(int d)

```

```

8  {
9
10     if(set[d] < 0)
11         return d;
12     return set[d] = set_find(set[d]);
13 }
14 int main(void)
15 {
16
17     int loop;
18     scanf("%d", &loop);
19     while(loop--)
20     {
21         scanf("%d%d", &n, &m);
22         memset(set, -1, sizeof(set));
23         for(int i=0; i<m; i++)
24         {
25             int a, b;
26             char s[5];
27             scanf("%s%d%d", s, &a, &b);
28             if(s[0] == 'A')
29             {
30                 if(set_find(a) != set_find(b) &&
31                     set_find(a) != set_find(b+n))
32                     printf("%s\n", "Not sure
33                         yet.");
34                 else if(set_find(a) ==
35                     set_find(b))
36                     printf("%s\n", "In the same
37                         gang.");
38                 else
39                     printf("%s\n", "In different
40                         gangs.");
41             }
42             else
43             {
44                 if(set_find(a) != set_find(b+n))
45                 {
46                     set[set_find(a)] =
47                         set_find(b+n);
48                     set[set_find(b)] =
49                         set_find(a+n);
50                 }
51             }
52         }
53     }
54     return 0;
55 }

```

vector 建樹找尋最近共同祖先

```

1  // #include <bits/stdc++.h>
2  #include <iostream>
3  #include <vector>
4  #include <cstring>
5  using namespace std;
6  const int N = 10000;

```

```
7 vector<int> a[N];
8 int f[N], r[N];
9 void DFS(int u, int dep)
10 {
11     r[u] = dep;
12     for(vector<int>::iterator it =
13         a[u].begin(); it != a[u].end(); it++)
14         DFS(*it, dep + 1);
15 }
16 int main()
17 {
18     int casenum, num, n, i, x, y;
19     scanf("%d", &casenum);
20     for(num=0; num<casenum; num++)
21     {
22         scanf("%d", &n);
23         for(i=0; i<n; i++) a[i].clear();
24         memset(f, 255, sizeof(f));
```

```
25         for(i=0; i<n-1; i++)
26         {
27             scanf("%d %d", &x, &y);
28             a[x-1].push_back(y-1);
29             f[y-1] = x-1;
30         }
31         for(i=0; f[i]>=0; i++);
32         DFS(i, 0);
33         scanf("%d %d", &x, &y);
34         x--; y--;
35         while(x != y)
36         {
37             if(r[x] > r[y]) x = f[x];
38             else y = f[y];
39         }
40         printf("%d\n", x+1);
41     }
42     return 0;
43 }
```