	٧.					
(\mathcal{C}	n	ıt.o	er	٦t.	S

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 €
      r[u] = dep;
11
      for(vector<int>::iterator it =
12
           a[u].begin(); it != a[u].end(); it++)
          DFS(*it, dep + 1);
13
14 }
15 int main()
16 {
17
18
       int casenum, num, n, i, x, y;
19
      scanf("%d", &casenum);
20
      for(num=0; num<casenum; num++)</pre>
21
      {
          scanf("%d", &n);
22
          for(i=0; i<n; i++) a[i].clear();</pre>
23
          memset(f, 255, sizeof(f));
24
          for(i=0; i<n-1; i++)</pre>
25
26
              scanf("%d %d", &x, &y);
27
              a[x-1].push_back(y-1);
28
              f[y-1] = x-1;
29
30
          for(i=0; f[i]>=0; i++);
31
          DFS(i, 0);
32
          scanf("%d %d", &x, &y);
          x--; y--;
34
          while(x != y)
35
36
              if(r[x] > r[y]) x = f[x];
37
38
              else y = f[y];
          printf("%d\n", x+1);
40
      }
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
      if(set[d] < 0)
10
          return d;
11
      return set[d] = set_find(set[d]);
13 }
14 int main(void)
15 {
16
17
      int loop;
      scanf("%d", &loop);
18
19
      while(loop--)
20
          scanf("%d%d", &n, &m);
21
22
          memset(set, -1, sizeof(set));
          for(int i=0; i<m; i++)</pre>
23
24
              int a, b;
25
26
              char s[5];
27
              scanf("%s%d%d", s, &a, &b);
              if(s[0] == 'A')
28
29
                  if(set_find(a) != set_find(b) &&
30
                      set_find(a) != set_find(b+n))
                     printf("%s\n", "Not sure
31
                          yet.");
                  else if(set_find(a) ==
                      set_find(b))
                     printf("%s\n", "In the same
33
                          gang.");
                  else
34
                     printf("%s\n", "In different
35
                          gangs.");
              }
36
              else
37
              {
38
                  if(set_find(a) != set_find(b+n))
39
40
                     set[set_find(a)] =
41
                          set_find(b+n);
42
                     set[set_find(b)] =
                          set_find(a+n);
                  }
43
              }
44
          }
45
46
      }
47
      return 0;
48 }
```

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