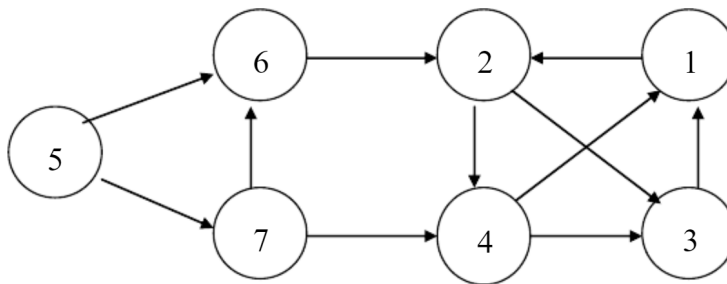


04/24/18 10:58:41 D:\git-repos\data-structure-homework\07\e20.cpp

```

1  #include <stdio>
2  #include <cstring>
3  #include <vector>
4  #define MXN 107
5  using namespace std;
6  vector<int> v[MXN];
7  bool tag[MXN];
8  int n, m;
9  int x;
10 bool dfs(int root, int dep, int target, int k) {
11     if (dep > k) return false;
12     if (target == root && dep == k) return true;
13     for (auto i : v[root]) {
14         if (!tag[i]) {
15             tag[i] = true;
16             if (dfs(i, dep + 1, target, k))
17                 return true;
18             tag[i] = false;
19         }
20     }
21     return false;
22 }
23 int main() {
24     scanf("%d %d", &n, &m);
25     for (int i = 0; i < m; ++i) {
26         int a, b;
27         scanf("%d %d", &a, &b);
28         v[a].push_back(b);
29         v[b].push_back(a);
30     }
31     int a, b, k;
32     while (~scanf("%d %d %d", &a, &b, &k)) {
33         memset(tag, 0, sizeof(tag));
34         tag[a] = true;
35         puts(dfs(a, 0, b, k) ? "YES" : "NO");
36     }
37 }
38 /**
39 root > ... > git-repos > data-structure-homework > 07 > g++ e20.cpp -std=c++11
40 root > ... > git-repos > data-structure-homework > 07 > ./a.out
41 7
42 11
43 5 6
44 5 7
45 7 6
46 6 2
47 7 4
48 2 4
49 2 3
50 4 1
51 1 2
52 3 1
53 4 3
54 1 4 5
55 YES
56 1 4 6
57 YES
58 1 4 7
59 NO
60 6 4 5
61 NO
62 6 4 3
63 YES
64 6 4 4
65 YES
66 ^Z
67 [1] + 1112 suspended ./a.out
68 root > ... > git-repos > data-structure-homework > 07 > 1 > ◀ 148 ◀ ? master
69 */

```



此例中本图为无向图

04/24/18 10:53:18 D:\git-repos\data-structure-homework\07\e21.cpp

```

1  #include <stdio>
2  #include <cstring>
3  #include <vector>
4  #include <set>
5  #include <map>
6  #define MXN 1007
7  using namespace std;
8  map<set<int>, vector<int>> sts;
9  vector<int> v[MXN];
10 int n, m;
11 int seq[MXN], loc[MXN];
12 char tag[MXN];
13 void dfs(int root, int dep) {
14     if (tag[root] == 1) {
15         set<int> tmp;
16         vector<int> tp;
17         for (int i = loc[root]; i < dep; ++i)
18             tmp.insert(seq[i]), tp.push_back(seq[i]);
19         if (sts.find(tmp) != sts.cend()) return;
20         sts.insert(make_pair(tmp, tp));
21         return;
22     }
23     tag[root] = 1;
24     seq[dep] = root;
25     loc[root] = dep;
26     for (auto i : v[root]) {
27         dfs(i, dep + 1);
28     }
29     tag[root] = 2;
30 }
31 int main() {
32     scanf("%d %d", &n, &m);
33     for (int i = 0; i < m; ++i) {
34         int a, b;
35         scanf("%d %d", &a, &b);
36         v[a].push_back(b);
37     }
38     for (int i = 1; i <= n; ++i)
39         if (tag[i] == 0)
40             dfs(i, 0);
41     printf("Found %d distinct simple cycle:\n", sts.size());
42     for (auto i : sts) {
43         for (auto j : i.second)
44             printf("%d ", j);
45         putchar('\n');
46     }
47 }
48 /**
49 root ► ... > git-repos > data-structure-homework > 07 ► g++ e21.cpp -std=c++11
50 root ► ... > git-repos > data-structure-homework > 07 ► ./a.out ◀ master
51 7
52 11
53 5 6
54 5 7
55 7 6
56 6 2
57 7 4
58 2 4
59 2 3
60 4 1
61 1 2
62 3 1
63 4 3
64 Found 3 distinct simple cycle:
65 1 2 3
66 1 2 4 3
67 1 2 4
68 root ► ... > git-repos > data-structure-homework > 07 ► ◀ master
69 */

```

