

그래프와 BFS

소스코드

최백준 choi@startlink.io

C++14

```
1 #include <cstdio>
2 #include <algorithm>
3 #include <cstring>
4 #include <vector>
5 #include <queue>
6 using namespace std;
7 vector<int> a[1001];
8 bool check[1001];
9 void dfs(int node) {
10     check[node] = true;
11     printf("%d ",node);
12     for (int i=0; i<a[node].size(); i++) {
13         int next = a[node][i];
14         if (check[next] == false) {
15             dfs(next);
16         }
17     }
18 }
19 void bfs(int start) {
20     queue<int> q;
21     memset(check,false,sizeof(check));
22     check[start] = true;
23     q.push(start);
24     while (!q.empty()) {
25         int node = q.front();
26         q.pop();
27         printf("%d ",node);
28         for (int i=0; i<a[node].size(); i++) {
29             int next = a[node][i];
30             if (check[next] == false) {
31                 check[next] = true;
32                 q.push(next);
33             }
34         }
35     }
36 }
37 int main() {
38     int n, m, start;
39     scanf("%d %d %d",&n,&m,&start);
40     for (int i=0; i<m; i++) {
41         int u,v;
42         scanf("%d %d",&u,&v);
43         a[u].push_back(v);
44         a[v].push_back(u);
45     }
46     for (int i=1; i<=n; i++) {
47         sort(a[i].begin(), a[i].end());
48     }
49     dfs(start);
50     puts("");
51     bfs(start);
52     puts("");
53     return 0;
54 }
```

결과	메모리	시간	코드 길이
맞았습니다!!!	1384 KB	4 ms	1175 B

Java

```
1 import java.util.*;
2
3 public class Main {
4     static ArrayList<Integer>[] a;
5     static boolean[] c;
6     public static void dfs(int x) {
7         if (c[x]) {
8             return;
9         }
10        c[x] = true;
11        System.out.print(x + " ");
12        for (int y : a[x]) {
13            if (c[y] == false) {
14                dfs(y);
15            }
16        }
17    }
18    public static void bfs(int start) {
19        Queue<Integer> q = new LinkedList<Integer>();
20        q.add(start);
21        c[start] = true;
22        while (!q.isEmpty()) {
23            int x = q.remove();
24            System.out.print(x + " ");
25            for (int y : a[x]) {
26                if (c[y] == false) {
27                    c[y] = true;
28                    q.add(y);
29                }
30            }
31        }
32    }
33    public static void main(String args[]) {
34        Scanner sc = new Scanner(System.in);
35        int n = sc.nextInt();
36        int m = sc.nextInt();
37        int start = sc.nextInt();
38        a = (ArrayList<Integer>[]) new ArrayList[n+1];
39        for (int i=1; i<=n; i++) {
40            a[i] = new ArrayList<Integer>();
41        }
42        for (int i=0; i<m; i++) {
43            int u = sc.nextInt();
44            int v = sc.nextInt();
45            a[u].add(v);
46            a[v].add(u);
47        }
48        for (int i=1; i<=n; i++) {
49            Collections.sort(a[i]);
50        }
51        c = new boolean[n+1];
52        dfs(start);
53        System.out.println();
54        c = new boolean[n+1];
55        bfs(start);
56        System.out.println();
57    }
58 }
```

결과	메모리	시간	코드 길이
맞았습니다!!!	33168 KB	444 ms	1521 B

C++14

```
1 #include <cstdio>
2 #include <algorithm>
3 #include <cstring>
4 #include <vector>
5 #include <queue>
6 using namespace std;
7 vector<int> a[20001];
8 int color[20001];
9 void dfs(int node, int c) {
10     color[node] = c;
11     for (int i=0; i<a[node].size(); i++) {
12         int next = a[node][i];
13         if (color[next] == 0) {
14             dfs(next, 3-c);
15         }
16     }
17 }
18 int main() {
19     int t;
20     scanf("%d\n",&t);
21     while (t--) {
22         int n, m;
23         scanf("%d %d",&n,&m);
24         for (int i=1; i<=n; i++) {
25             a[i].clear();
26             color[i] = 0;
27         }
28         for (int i=0; i<m; i++) {
29             int u,v;
30             scanf("%d %d",&u,&v);
31             a[u].push_back(v);
32             a[v].push_back(u);
33         }
34         for (int i=1; i<=n; i++) {
35             if (color[i] == 0) {
36                 dfs(i, 1);
37             }
38         }
39         bool ok = true;
40         for (int i=1; i<=n; i++) {
41             for (int k=0; k<a[i].size(); k++) {
42                 int j = a[i][k];
43                 if (color[i] == color[j]) {
44                     ok = false;
45                 }
46             }
47         }
48         printf("%s\n",ok ? "YES" : "NO");
49     }
50     return 0;
51 }
```

0: 방문 X
1: 그룹 A
2: 그룹 B

C → 3-C
1 → 2
2 → 1

결과	메모리	시간	코드 길이
맞았습니다!!	10428 KB	384 ms	1161 B

C++17

```
1 #include <cstdio>
2 #include <algorithm>
3 #include <cstring>
4 #include <vector>
5 #include <queue>
6 using namespace std;
7 vector<int> a[20001];
8 int color[20001];
9 bool dfs(int node, int c) {
10     color[node] = c;
11     for (int i=0; i<a[node].size(); i++) {
12         int next = a[node][i];
13         if (color[next] == 0) {
14             if (dfs(next, 3-c) == false) {
15                 return false;
16             }
17         } else if (color[next] == color[node]) {
18             return false;
19         }
20     }
21     return true;
22 }
23 int main() {
24     int t;
25     scanf("%d\n",&t);
26     while (t--) {
27         int n, m;
28         scanf("%d %d",&n,&m);
29         for (int i=1; i<=n; i++) {
30             a[i].clear();
31             color[i] = 0;
32         }
33         for (int i=0; i<m; i++) {
34             int u,v;
35             scanf("%d %d",&u,&v);
36             a[u].push_back(v);
37             a[v].push_back(u);
38         }
39         bool ok = true;
40         for (int i=1; i<=n; i++) {
41             if (color[i] == 0) {
42                 if (dfs(i, 1) == false) {
43                     ok = false;
44                 }
45             }
46         }
47         printf("%s\n",ok ? "YES" : "NO");
48     }
49     return 0;
50 }
```

←

ok = false

결과	메모리	시간	코드 길이
맞았습니다!!	10428 KB	328 ms	1147 B

Java

```
1 import java.util.*;
2
3 public class Main {
4     public static void dfs(ArrayList<Integer>[] a, int[] color, int x, int c) {
5         color[x] = c;
6         for (int y : a[x]) {
7             if (color[y] == 0) {
8                 dfs(a, color, y, 3-c);
9             }
10        }
11    }
12
13    public static void main(String args[]) {
14        Scanner sc = new Scanner(System.in);
15        int t = sc.nextInt();
16        while (t-- > 0) {
17            int n = sc.nextInt();
18            int m = sc.nextInt();
19            ArrayList<Integer>[] a = (ArrayList<Integer>[]) new ArrayList[n+1];
20            for (int i=1; i<=n; i++) {
21                a[i] = new ArrayList<Integer>();
22            }
23            for (int i=0; i<m; i++) {
24                int u = sc.nextInt();
25                int v = sc.nextInt();
26                a[u].add(v);
27                a[v].add(u);
28            }
29            int[] color = new int[n+1];
30            boolean ok = true;
31            for (int i=1; i<=n; i++) {
32                if (color[i] == 0) {
33                    dfs(a, color, i, 1);
34                }
35            }
36            for (int i=1; i<=n; i++) {
37                for (int j : a[i]) {
38                    if (color[i] == color[j]) {
39                        ok = false;
40                    }
41                }
42            }
43            if (ok) {
44                System.out.println("YES");
45            } else {
46                System.out.println("NO");
47            }
48        }
49    }
50 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	191784 KB	2448 ms	1450 B

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <cstring>
5 using namespace std;
6 int d[1001][1001];
7 int main() {
8     int n;
9     cin >> n;
10    memset(d,-1,sizeof(d));
11    queue<pair<int,int>> q;
12    q.push(make_pair(1,0));
13    d[1][0] = 0;
14    while (!q.empty()) {
15        int s,c;
16        tie(s, c) = q.front();
17        q.pop();
18        if (d[s][s] == -1) {
19            d[s][s] = d[s][c] + 1;
20            q.push(make_pair(s,s));
21        }
22        if (s+c <= n && d[s+c][c] == -1) {
23            d[s+c][c] = d[s][c] + 1;
24            q.push(make_pair(s+c, c));
25        }
26        if (s-1 >= 0 && d[s-1][c] == -1) {
27            d[s-1][c] = d[s][c] + 1;
28            q.push(make_pair(s-1, c));
29        }
30    }
31    int ans = -1;
32    for (int i=0; i<=n; i++) {
33        if (d[n][i] != -1) {
34            if (ans == -1 || ans > d[n][i]) {
35                ans = d[n][i];
36            }
37        }
38    }
39    cout << ans << '\n';
40    return 0;
41 }
```

Handwritten notes for C++ solution:

- check dist
- $|x_2| \geq 0$
- t, f with -1
- $D[N]$, $D[M]$
- N , $?$, $0 \sim N$

결과	메모리	시간	코드 길이
맞았습니다!!	5900 KB	16 ms	942 B

Java

```
1 import java.util.*;
2 public class Main {
3     public static void main(String args[]) {
4         Scanner sc = new Scanner(System.in);
5         int n = sc.nextInt();
6         int[][] d = new int[n+1][n+1];
7         for (int i=0; i<=n; i++) {
8             Arrays.fill(d[i], -1);
9         }
10        Queue<Integer> q = new LinkedList<Integer>();
11        q.add(1);
12        q.add(0);
13        d[1][0] = 0;
14        while (!q.isEmpty()) {
15            int s = q.remove();
16            int c = q.remove();
17            if (d[s][s] == -1) {
18                d[s][s] = d[s][c] + 1;
19                q.add(s); q.add(s);
20            }
21            if (s+c <= n && d[s+c][c] == -1) {
22                d[s+c][c] = d[s][c] + 1;
23                q.add(s+c); q.add(c);
24            }
25            if (s-1 >= 0 && d[s-1][c] == -1) {
26                d[s-1][c] = d[s][c] + 1;
27                q.add(s-1); q.add(c);
28            }
29        }
30        int ans = -1;
31        for (int i=0; i<=n; i++) {
32            if (d[n][i] != -1) {
33                if (ans == -1 || ans > d[n][i]) {
34                    ans = d[n][i];
35                }
36            }
37        }
38        System.out.println(ans);
39    }
40 }
```

Handwritten notes for Java solution:

- S, C
- Pair, tuple
- SCSCSC...

결과	메모리	시간	코드 길이
맞았습니다!!	55628 KB	268 ms	1154 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <deque>
4 using namespace std;
5 bool c[1000000];
6 int d[1000000];
7 int MAX = 1000000;
8 int main() {
9     int n, m;
10    cin >> n >> m;
11    c[n] = true;
12    d[n] = 0;
13    queue<int> q;
14    queue<int> next_queue;
15    q.push(n);
16    while (!q.empty()) {
17        int now = q.front();
18        q.pop();
19        if (now*2 < MAX) {
20            if (c[now*2] == false) {
21                q.push(now*2);
22                c[now*2] = true;
23                d[now*2] = d[now];
24            }
25        }
26        if (now-1 >= 0) {
27            if (c[now-1] == false) {
28                next_queue.push(now-1);
29                c[now-1] = true;
30                d[now-1] = d[now] + 1;
31            }
32        }
33        if (now+1 < MAX) {
34            if (c[now+1] == false) {
35                next_queue.push(now+1);
36                c[now+1] = true;
37                d[now+1] = d[now] + 1;
38            }
39        }
40        if (q.empty()) {
41            q = next_queue;
42            next_queue = queue<int>();
43        }
44    }
45    cout << d[m] << '\n';
46    return 0;
47 }
```

거리 같

(+1)

결과	메모리	시간	코드 길이
맞았습니다!!	8072 KB	28 ms	1070 B

Java

```
1 import java.util.*;
2
3 public class Main {
4     public static final int MAX = 1000000;
5     public static void main(String args[]) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         int m = sc.nextInt();
9         boolean[] c = new boolean[MAX];
10        int[] d = new int[MAX];
11        c[n] = true;
12        d[n] = 0;
13        Queue<Integer> q = new LinkedList<Integer>();
14        Queue<Integer> next_queue = new LinkedList<Integer>();
15        q.add(n);
16        while (!q.isEmpty()) {
17            int now = q.remove();
18            for (int next : new int[]{now*2, now-1, now+1}) {
19                if (next >= 0 && next < MAX) {
20                    if (c[next] == false) {
21                        c[next] = true;
22                        if (now*2 == next) {
23                            q.add(next);
24                            d[next] = d[now];
25                        } else {
26                            next_queue.add(next);
27                            d[next] = d[now] + 1;
28                        }
29                    }
30                }
31            }
32            if (q.isEmpty()) {
33                q = next_queue;
34                next_queue = new LinkedList<Integer>();
35            }
36        }
37        System.out.println(d[m]);
38    }
39 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	63644 KB	332 ms	1289 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <deque>
4 using namespace std;
5 bool c[1000000];
6 int d[1000000];
7 int MAX = 1000000;
8 int main() {
9     int n, m;
10    cin >> n >> m;
11    c[n] = true;
12    d[n] = 0;
13    deque<int> q;
14    q.push_back(n);
15    while (!q.empty()) {
16        int now = q.front();
17        q.pop_front();
18        if (now*2 < MAX) {
19            if (c[now*2] == false) {
20                q.push_front(now*2);
21                c[now*2] = true;
22                d[now*2] = d[now];
23            }
24        }
25        if (now-1 >= 0) {
26            if (c[now-1] == false) {
27                q.push_back(now-1);
28                c[now-1] = true;
29                d[now-1] = d[now] + 1;
30            }
31        }
32        if (now+1 < MAX) {
33            if (c[now+1] == false) {
34                q.push_back(now+1);
35                c[now+1] = true;
36                d[now+1] = d[now] + 1;
37            }
38        }
39    }
40    cout << d[m] << '\n';
41    return 0;
42 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	7532 KB	12 ms	950 B

Java

```
1 import java.util.*;
2
3 public class Main {
4     public static final int MAX = 1000000;
5     public static void main(String args[]) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         int m = sc.nextInt();
9         boolean[] c = new boolean[MAX];
10        int[] d = new int[MAX];
11        c[n] = true;
12        d[n] = 0;
13        ArrayDeque<Integer> q = new ArrayDeque<Integer>();
14        q.add(n);
15        while (!q.isEmpty()) {
16            int now = q.poll();
17            for (int next : new int[]{now*2, now-1, now+1}) {
18                if (next >= 0 && next < MAX) {
19                    if (c[next] == false) {
20                        c[next] = true;
21                        if (next == now*2) {
22                            q.addFirst(next);
23                            d[next] = d[now];
24                        } else {
25                            q.addLast(next);
26                            d[next] = d[now] + 1;
27                        }
28                    }
29                }
30            }
31        }
32        System.out.println(d[m]);
33    }
34 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	54148 KB	284 ms	1076 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <cstdio>
4 #include <tuple>
5 using namespace std;
6 int a[1000][1000];
7 int d[1000][1000][2];
8 int dx[] = {0, 0, 1, -1};
9 int dy[] = {1, -1, 0, 0};
10 int main() {
11     int n, m;
12     scanf("%d %d", &n, &m);
13     for (int i=0; i<n; i++) {
14         for (int j=0; j<m; j++) {
15             scanf("%1d", &a[i][j]);
16         }
17     }
18     queue<tuple<int, int, int>> q;
19     d[0][0][0] = 1;
20     q.push(make_tuple(0, 0, 0));
21     while (!q.empty()) {
22         int x, y, z;
23         tie(x, y, z) = q.front(); q.pop();
24         for (int k=0; k<4; k++) {
25             int nx = x+dx[k];
26             int ny = y+dy[k];
27             if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
28             if (a[nx][ny] == 0 && d[nx][ny][z] == 0) {
29                 d[nx][ny][z] = d[x][y][z] + 1;
30                 q.push(make_tuple(nx, ny, z));
31             }
32             if (z == 0 && a[nx][ny] == 1 && d[nx][ny][z+1] == 0) {
33                 d[nx][ny][z+1] = d[x][y][z] + 1;
34                 q.push(make_tuple(nx, ny, z+1));
35             }
36         }
37     }
38     if (d[n-1][m-1][0] != 0 && d[n-1][m-1][1] != 0) {
39         cout << min(d[n-1][m-1][0], d[n-1][m-1][1]);
40     } else if (d[n-1][m-1][0] != 0) {
41         cout << d[n-1][m-1][0];
42     } else if (d[n-1][m-1][1] != 0) {
43         cout << d[n-1][m-1][1];
44     } else {
45         cout << -1;
46     }
47     cout << '\n';
48     return 0;
49 }
```

(x, y) → (nx, ny)
② → z+1

if (d[n-1][m-1][0] != 0 && d[n-1][m-1][1] != 0) {
 cout << min(d[n-1][m-1][0], d[n-1][m-1][1]);
} else if (d[n-1][m-1][0] != 0) {
 cout << d[n-1][m-1][0];
} else if (d[n-1][m-1][1] != 0) {
 cout << d[n-1][m-1][1];
} else {
 cout << -1;
}

결과	메모리	시간	코드 길이
맞았습니다!!	13708 KB	116 ms	1358 B

Java

```
1 import java.util.*;
2 class Pair {
3     int x, y, z;
4     Pair(int x, int y, int z) {
5         this.x = x;
6         this.y = y;
7         this.z = z;
8     }
9 }
10 public class Main {
11     public static int[] dx = {1, -1, 0, 0};
12     public static int[] dy = {0, 0, 1, -1};
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         int n = sc.nextInt();
16         int m = sc.nextInt();
17         sc.nextLine();
18         int[][] a = new int[n][m];
19         int[][][] d = new int[n][m][2];
20         for (int i=0; i<n; i++) {
21             String s = sc.nextLine();
22             for (int j=0; j<m; j++) {
23                 a[i][j] = s.charAt(j) - '0';
24             }
25         }
26         d[0][0][0] = 1;
27         Queue<Pair> q = new LinkedList<Pair>();
28         q.offer(new Pair(0, 0, 0));
29         while (!q.isEmpty()) {
30             Pair p = q.remove();
31             int x = p.x;
32             int y = p.y;
33             int z = p.z;
34             for (int k=0; k<4; k++) {
35                 int nx = x+dx[k];
36                 int ny = y+dy[k];
37                 if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
38                 if (a[nx][ny] == 0 && d[nx][ny][z] == 0) {
39                     d[nx][ny][z] = d[x][y][z] + 1;
40                     q.offer(new Pair(nx, ny, z));
41                 }
42                 if (z == 0 && a[nx][ny] == 1 && d[nx][ny][z+1] == 0) {
43                     d[nx][ny][z+1] = d[x][y][z] + 1;
44                     q.offer(new Pair(nx, ny, z+1));
45                 }
46             }
47         }
48         if (d[n-1][m-1][0] != 0 && d[n-1][m-1][1] != 0) {
49             System.out.println(Math.min(d[n-1][m-1][0], d[n-1][m-1][1]));
50         } else if (d[n-1][m-1][0] != 0) {
51             System.out.println(d[n-1][m-1][0]);
52         } else if (d[n-1][m-1][1] != 0) {
53             System.out.println(d[n-1][m-1][1]);
54         } else {
55             System.out.println(-1);
56         }
57     }
58 }
59 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	95336 KB	828 ms	1887 B

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.