

BFS (연습2)

소스코드

최백준 choi@startlink.io

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 using namespace std;
5 int dx[] = {0,0,1,-1};
6 int dy[] = {1,-1,0,0};
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    vector<string> a(n);
11    for (int i=0; i<n; i++) {
12        cin >> a[i];
13    }
14    bool ok = true;
15    for (int i=0; i<n; i++) {
16        for (int j=0; j<m; j++) {
17            if (a[i][j] == 'S') {
18                for (int k=0; k<4; k++) {
19                    int x = i+dx[k];
20                    int y = j+dy[k];
21                    if (0 <= x && x < n && 0 <= y && y < m) {
22                        if (a[x][y] == 'W') {
23                            ok = false;
24                        }
25                    }
26                }
27            }
28        }
29    }
30    if (!ok) {
31        cout << 0 << '\n';
32    } else {
33        cout << 1 << '\n';
34        for (int i=0; i<n; i++) {
35            for (int j=0; j<m; j++) {
36                if (a[i][j] == '.') {
37                    cout << 'D';
38                } else {
39                    cout << a[i][j];
40                }
41            }
42            cout << '\n';
43        }
44    }
45    return 0;
46 }
47
```

결과	메모리	시간	코드 길이
맞았습니다!!	2384 KB	20 ms	1106 B

C++14

```

1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 int dist[1000001]; → 거리 (버튼 횟수)
5 bool check[1000001]; →
6 int main() {
7     int f,s,g,u,d;
8     cin >> f >> s >> g >> u >> d;
9     queue<int> q;
10    q.push(s);
11    check[s] = true;
12    while (!q.empty()) {
13        int now = q.front();
14        q.pop();
15        if (now + u <= f && check[now+u] == false) {
16            dist[now+u] = dist[now] + 1;
17            check[now+u] = true;
18            q.push(now+u);
19        }
20        if (now - d >= 1 && check[now-d] == false) {
21            dist[now-d] = dist[now] + 1;
22            check[now-d] = true;
23            q.push(now-d);
24        }
25    }
26    if (check[g]) {
27        cout << dist[g] << '\n';
28    } else {
29        cout << "use the stairs\n";
30    }
31    return 0;
32 }

```

Handwritten notes and annotations:

- Handwritten "거리 (버튼 횟수)" with an arrow pointing to the `dist` array.
- Handwritten "1-F" in a circle next to the `while` loop.
- Handwritten "O(F)" next to the `while` loop.
- Handwritten "use the stairs\n" in red and circled in the `else` block.

결과

메모리

시간

코드 길이

맞았습니다!!

6872 KB

8 ms

746 B

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <algorithm>
5 #include <deque>
6 #include <tuple>
7 using namespace std;
8 int dx[] = {0, 0, 1, -1};
9 int dy[] = {1, -1, 0, 0};
10 vector<vector<int>> bfs(vector<string> &a, int x, int y) {
11     int n = a.size();
12     int m = a[0].size();
13     vector<vector<int>> d(n, vector<int>(m));
14     for (int i=0; i<n; i++) {
15         for (int j=0; j<m; j++) {
16             d[i][j] = -1;
17         }
18     }
19     deque<pair<int,int>> q;
20     q.push_back(make_pair(x, y));
21     d[x][y] = 0;
22     while (!q.empty()) {
23         tie(x,y) = q.front(); q.pop_front();
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 (d[nx][ny] != -1) continue;
29             if (a[nx][ny] == '*') continue;
30             if (a[nx][ny] == '#') {
31                 d[nx][ny] = d[x][y] + 1;
32                 q.push_back(make_pair(nx,ny));
33             } else {
34                 d[nx][ny] = d[x][y];
35                 q.push_front(make_pair(nx,ny));
36             }
37         }
38     }
39     return d;
40 }
41 int main() {
42     int t;
43     cin >> t;
44     while (t--) {
45         int n, m;
46         cin >> n >> m;
47         vector<string> a(n+2);
48         for (int i=1; i<=n; i++) {
49             cin >> a[i];
50             a[i] = "." + a[i] + ".";
51         }
52         n += 2;
53         m += 2;
54         for (int j=0; j<m; j++) {
55             a[0] += ".";
56             a[n-1] += ".";
57         }
58         vector<vector<int>> d0 = bfs(a, 0, 0);
59         int x1, y1, x2, y2;
60         x1 = y1 = x2 = y2 = -1;
61         for (int i=0; i<n; i++) {
62             for (int j=0; j<m; j++) {
63                 if (a[i][j] == '$') {
64                     if (x1 == -1) {
65                         x1 = i;
66                         y1 = j;
67                     } else {
68                         x2 = i;
69                         y2 = j;
70                     }
71                 }
72             }
73         }
74         vector<vector<int>> d1 = bfs(a, x1, y1);
75         vector<vector<int>> d2 = bfs(a, x2, y2);
76         int ans = n*m;
77         for (int i=0; i<n; i++) {
78             for (int j=0; j<m; j++) {
79                 if (a[i][j] == '*') continue;
80                 int cur = d0[i][j] + d1[i][j] + d2[i][j];
81                 if (a[i][j] == '#') cur -= 2;
82                 if (ans > cur) ans = cur;
83             }
84         }
85         cout << ans << '\n';
86     }
87     return 0;
88 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

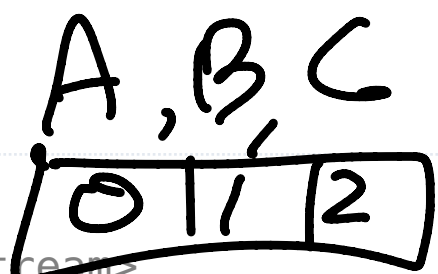
2136 KB

16 ms

2410 B

C++14

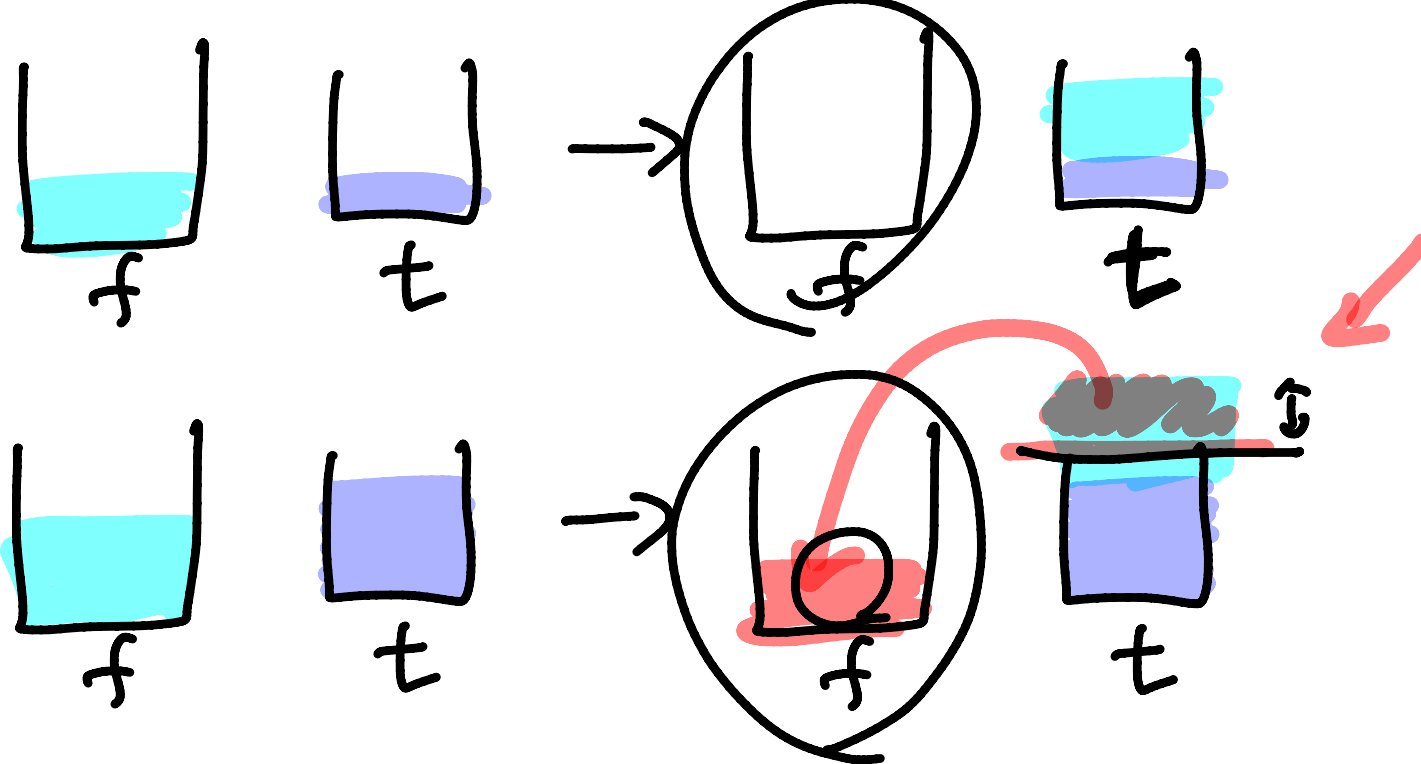
```
1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 bool ans[201];
5 bool check[201][201];
6 int cap[3];
7 int from[] = {0, 0, 1, 1, 2, 2};
8 int to[] = {1, 2, 0, 2, 0, 1};
9 int main() {
10     for (int i=0; i<3; i++) {
11         cin >> cap[i];
12     }
13     int sum = cap[2];
14     queue<pair<int,int>> q;
15     q.push(make_pair(0, 0));
16     check[0][0] = true;
17     ans[cap[2]] = true;
18     while (!q.empty()) {
19         int cur[3];
20         cur[0] = q.front().first;
21         cur[1] = q.front().second;
22         cur[2] = sum - cur[0] - cur[1];
23         q.pop();
24         for (int k=0; k<6; k++) {
25             int next[3] = {cur[0], cur[1], cur[2]};
26             next[to[k]] += next[from[k]];
27             next[from[k]] = 0;
28             if (next[to[k]] >= cap[to[k]]) {
29                 next[from[k]] = next[to[k]] - cap[to[k]];
30                 next[to[k]] = cap[to[k]];
31             }
32             if (!check[next[0]][next[1]]) {
33                 check[next[0]][next[1]] = true;
34                 q.push(make_pair(next[0], next[1]));
35                 if (next[0] == 0) {
36                     ans[next[2]] = true;
37                 }
38             }
39         }
40     }
41     for (int i=0; i<=cap[2]; i++) {
42         if (ans[i]) {
43             cout << i << ' ';
44         }
45     }
46     cout << '\n';
47     return 0;
48 }
```



ans[] = T/F
A가 비어있을 때
이 물이 T/F만큼
들어가는 것이
가능 / 불가능

6개씩
0, 1 f → t
{0, 0}

f → t



결과

맞았습니다!!

메모리

2028 KB

시간

0 ms

코드 길이

1266 B

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <algorithm>
5 using namespace std;
6 int n, m;
7 int a[1000][1000];
8 int group[1000][1000];
9 int group_size[1000*1000];
10 int groupn = 0;
11 int dx[] = {0,0,1,-1};
12 int dy[] = {1,-1,0,0};
13 void bfs(int sx, int sy) {
14     groupn += 1;
15     queue<pair<int,int>> q;
16     q.push(make_pair(sx,sy));
17     group[sx][sy] = groupn;
18     int cnt = 1;
19     while (!q.empty()) {
20         int x, y;
21         tie(x,y) = q.front(); q.pop();
22         for (int k=0; k<4; k++) {
23             int nx = x+dx[k];
24             int ny = y+dy[k];
25             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
26                 if (group[nx][ny] == 0 && a[nx][ny] == 1) {
27                     group[nx][ny] = groupn;
28                     q.push(make_pair(nx,ny));
29                     cnt += 1;
30                 }
31             }
32         }
33     }
34     group_size[groupn] = cnt;
35 }
36 int main() {
37     cin >> n >> m;
38     for (int i=0; i<n; i++) {
39         for (int j=0; j<m; j++) {
40             cin >> a[i][j];
41         }
42     }
43     for (int i=0; i<n; i++) {
44         for (int j=0; j<m; j++) {
45             if (a[i][j] == 1 && group[i][j] == 0) {
46                 bfs(i, j);
47             }
48         }
49     }
50     int ans = 0;
51     for (int i=0; i<n; i++) {
52         for (int j=0; j<m; j++) {
53             if (a[i][j] == 0) {
54                 vector<int> near;
55                 for (int k=0; k<4; k++) {
56                     int nx = i+dx[k];
57                     int ny = j+dy[k];
58                     if (0 <= nx && nx < n && 0 <= ny && ny < m) {
59                         if (a[nx][ny] == 1) {
60                             near.push_back(group[nx][ny]);
61                         }
62                     }
63                 }
64                 sort(near.begin(), near.end());
65                 near.erase(unique(near.begin(), near.end()), near.end());
66                 int sum = 1;
67                 for (int neighbor : near) {
68                     sum += group_size[neighbor];
69                 }
70                 if (ans < sum) ans = sum;
71             }
72         }
73     }
74     cout << ans << '\n';
75     return 0;
76 }
77
```

그룹 = 모양

그룹의 크기

그룹의 개수

① 연산

민감하다

3점 /

그룹끼리

O(NM)

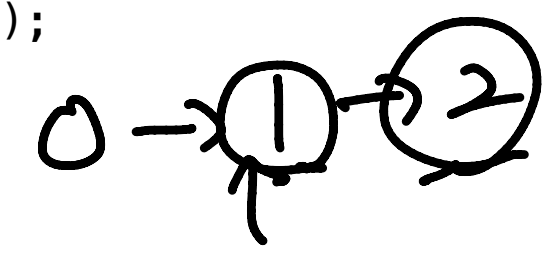
C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <cstring>
5 using namespace std;
6 int dx[] = {0,0,1,-1,-2,-1,1,2,2,1,-1,-2};
7 int dy[] = {1,-1,0,0,1,2,2,1,-1,-2,-2,-1};
8 int cost[] = {0,0,0,0,1,1,1,1,1,1,1,1};
9 int a[200][200];
10 int d[200][200][31];
11 int main() {
12     int l;
13     cin >> l;
14     int n, m;
15     cin >> m >> n;
16     for (int i=0; i<n; i++) {
17         for (int j=0; j<m; j++) {
18             cin >> a[i][j];
19         }
20     }
21     memset(d,-1,sizeof(d));
22     queue<tuple<int,int,int>> q;
23     d[0][0][0] = 0;
24     q.push(make_tuple(0,0,0));
25     while (!q.empty()) {
26         int x, y, c;
27         tie(x,y,c) = q.front();
28         q.pop();
29         for (int k=0; k<12; k++) {
30             int nx = x+dx[k];
31             int ny = y+dy[k];
32             int nc = c+cost[k];
33             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
34                 if (a[nx][ny] == 1) continue;
35                 if (nc <= l) {
36                     if (d[nx][ny][nc] == -1) {
37                         d[nx][ny][nc] = d[x][y][c] + 1;
38                         q.push(make_tuple(nx,ny,nc));
39                     }
40                 }
41             }
42         }
43     }
44     int ans = -1;
45     for (int i=0; i<=l; i++) {
46         if (d[n-1][m-1][i] == -1) continue;
47         if (ans == -1 || ans > d[n-1][m-1][i]) {
48             ans = d[n-1][m-1][i];
49         }
50     }
51     cout << ans << '\n';
52     return 0;
53 }
```

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 using namespace std;
5 int a[55][55];
6 int d[55][55];
7 int n, m;
8 int dx[] = {0,0,1,-1,1,1,-1,-1};
9 int dy[] = {1,-1,0,0,1,-1,1,-1};
10 int go(int sx, int sy) {
11     for (int i=0; i<n; i++) {
12         for (int j=0; j<m; j++) {
13             d[i][j] = -1;
14         }
15     }
16     d[sx][sy] = 0;
17     queue<pair<int,int>> q;
18     q.push(make_pair(sx,sy));
19     while (!q.empty()) {
20         int x, y;
21         tie(x, y) = q.front(); q.pop();
22         for (int k=0; k<8; k++) {
23             int nx = x+dx[k];
24             int ny = y+dy[k];
25             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
26                 if (d[nx][ny] == -1) {
27                     if (a[nx][ny] == 1) {
28                         return d[x][y]+1;
29                     } else {
30                         q.push(make_pair(nx,ny));
31                         d[nx][ny] = d[x][y] + 1;
32                     }
33                 }
34             }
35         }
36     }
37     return 0;
38 }
39 int main() {
40     cin >> n >> m;
41     for (int i=0; i<n; i++) {
42         for (int j=0; j<m; j++) {
43             cin >> a[i][j];
44         }
45     }
46     int ans = 0;
47     for (int i=0; i<n; i++) {
48         for (int j=0; j<m; j++) {
49             if (a[i][j] == 0) {
50                 int dist = go(i, j);
51                 if (ans < dist) ans = dist;
52             }
53         }
54     }
55     cout << ans << '\n';
56     return 0;
57 }
```

50x50
8개의 방향
(sx, sy) 부터
3단계까지의
거리
(한칸 거리)



이동
산의 .

변환

) NM
NM
 $O(NM)^2$

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <tuple>
4 #include <queue>
5 #include <string>
6 #include <vector>
7 using namespace std;
8 int dx[] = {0,0,1,-1};
9 int dy[] = {1,-1,0,0};
10 vector<vector<int>> bfs(vector<string> &a, int sx, int sy) {
11     int n = a.size();
12     int m = a[0].size();
13     vector<vector<int>> dist(n, vector<int>(m,-1));
14     queue<pair<int,int>> q;
15     q.push(make_pair(sx,sy));
16     dist[sx][sy] = 0;
17     while (!q.empty()) {
18         int x, y;
19         tie(x,y) = q.front(); q.pop();
20         for (int k=0; k<4; k++) {
21             int nx = x+dx[k];
22             int ny = y+dy[k];
23             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
24                 if (dist[nx][ny] == -1 && a[nx][ny] != 'x') {
25                     dist[nx][ny] = dist[x][y] + 1;
26                     q.push(make_pair(nx,ny));
27                 }
28             }
29         }
30     }
31     return dist;
32 }
33 int main() {
34     while (true) {
35         int n, m;
36         cin >> m >> n;
37         if (n == 0 && m == 0) break;
38         vector<string> a(n);
39         for (int i=0; i<n; i++) {
40             cin >> a[i];
41         }
42         vector<pair<int,int>> b(1);
43         for (int i=0; i<n; i++) {
44             for (int j=0; j<m; j++) {
45                 if (a[i][j] == 'o') {
46                     b[0] = make_pair(i,j);
47                 } else if (a[i][j] == '*') {
48                     b.push_back(make_pair(i,j));
49                 }
50             }
51         }
52         int l = b.size();
53         vector<vector<int>> d(l, vector<int>(l));
54         bool ok = true;
55         for (int i=0; i<l; i++) {
56             auto dist = bfs(a,b[i].first,b[i].second);
57             for (int j=0; j<l; j++) {
58                 d[i][j] = dist[b[j].first][b[j].second];
59                 if (d[i][j] == -1) {
60                     ok = false;
61                 }
62             }
63         }
64         if (ok == false) {
65             cout << -1 << '\n';
66             continue;
67         }
68         vector<int> p(l-1);
69         for (int i=0; i<l-1; i++) {
70             p[i] = i+1;
71         }
72         int ans = -1;
73         do {
74             int now = d[0][p[0]];
75             for (int i=0; i<l-2; i++) {
76                 now += d[p[i]][p[i+1]];
77             }
78             if (ans == -1 || ans > now) {
79                 ans = now;
80             }
81         } while(next_permutation(p.begin(), p.end()));
82         cout << ans << '\n';
83     }
84     return 0;
85 }
```

BFS

(sx,sy) 시작
점

(0,0) →

b[0] =

가능한 모든 경우
구하기

d[0][0] ≠ 1번 → 5번
거리

↙

10! x 10

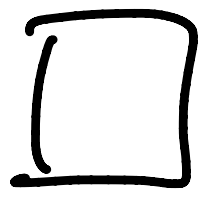
BFS
필요 없음

10! x NM

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4 using namespace std;
5 int dx[] = {0,0,1,-1};
6 int dy[] = {1,-1,0,0};
7 int main() {
8     int n;
9     cin >> n;
10    vector<string> s(n);
11    vector<vector<int>> b(n, vector<int>(n));
12    vector<pair<int,int>> v;
13    int start=-1, end=-1;
14    for (int i=0; i<n; i++) {
15        cin >> s[i];
16        for (int j=0; j<n; j++) {
17            if (s[i][j] == '#') {
18                if (start == -1) {
19                    start = v.size();
20                } else {
21                    end = v.size();
22                }
23                v.push_back(make_pair(i,j));
24                b[i][j] = v.size()-1;
25            } else if (s[i][j] == '!') {
26                v.push_back(make_pair(i,j));
27                b[i][j] = v.size()-1;
28            }
29        }
30    }
31    int m = v.size();
32    vector<vector<bool>> a(m, vector<bool>(m, false));
33    for (int i=0; i<m; i++) {
34        for (int k=0; k<4; k++) {
35            int x = v[i].first+dx[k];
36            int y = v[i].second+dy[k];
37            while (0 <= x && x < n && 0 <= y && y < n) {
38                if (s[x][y] == '*') break;
39                if (s[x][y] == '!' || s[x][y] == '#') {
40                    a[i][b[x][y]] = true;
41                }
42                x += dx[k];
43                y += dy[k];
44            }
45        }
46    }
47    queue<int> q;
48    vector<int> dist(m, -1);
49    q.push(start);
50    dist[start] = 0;
51    while (!q.empty()) {
52        int now = q.front();
53        q.pop();
54        for (int i=0; i<m; i++) {
55            if (a[now][i] && dist[i] == -1) {
56                dist[i] = dist[now]+1;
57                q.push(i);
58            }
59        }
60    }
61    cout << dist[end]-1 << '\n';
62    return 0;
63 }
64
```

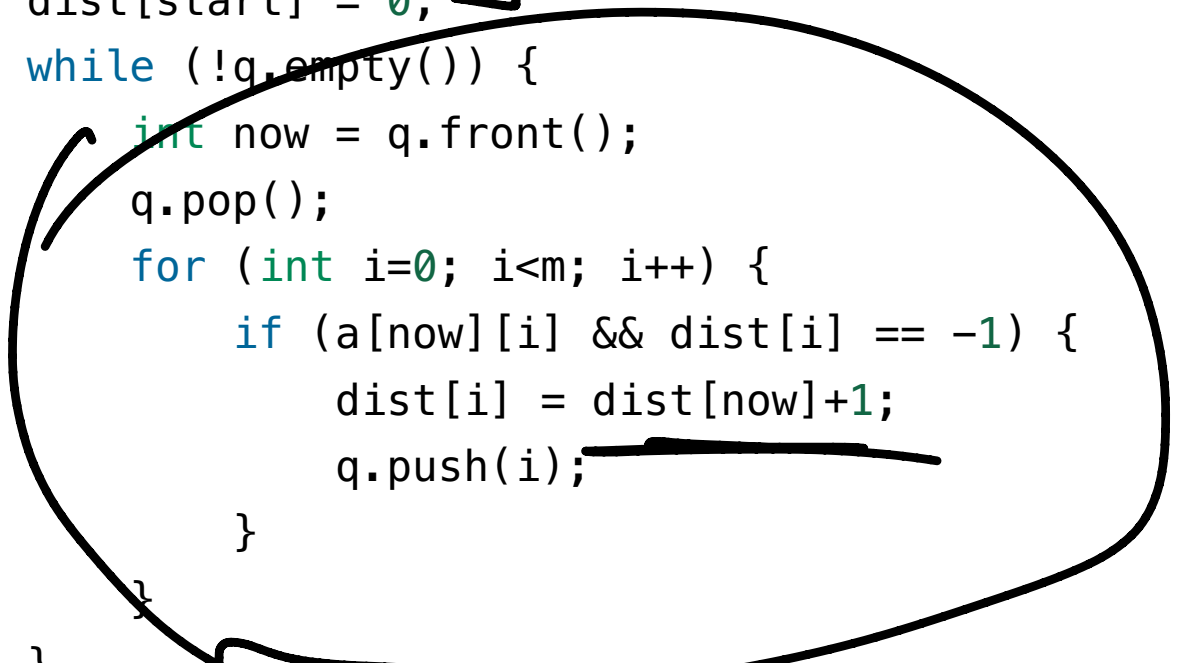
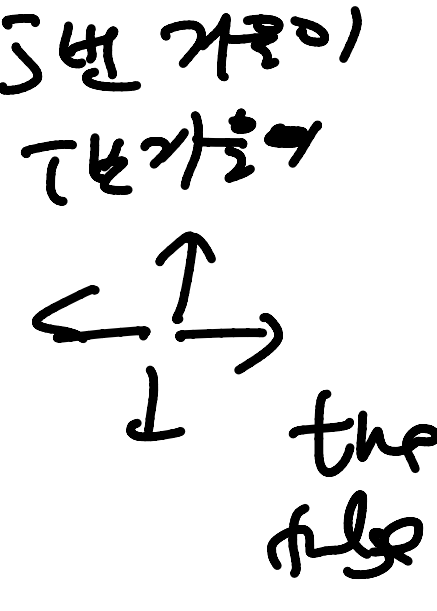
4



b[0][0] ~ b[n-1][n-1] 네 거울
거울의 번호

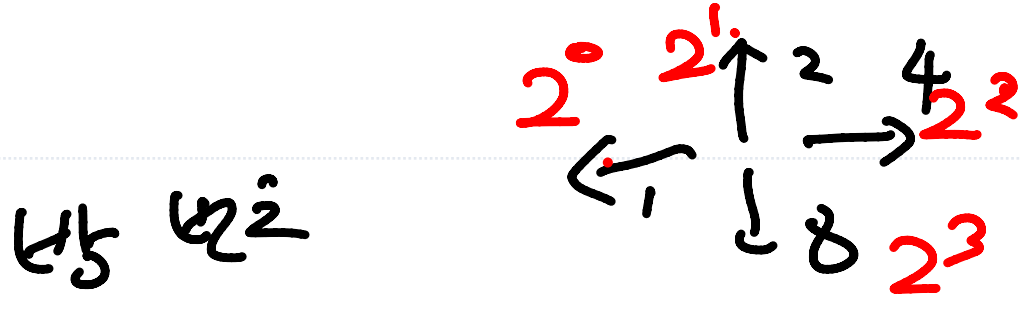
v[i] =
거울의
위치

a[i][j] = ① → ⑤



C++14

```
1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 int n, m;
5 int a[50][50];
6 int d[50][50];
7 int room[50*50];
8 int dx[] = {0,-1,0,1};
9 int dy[] = {-1,0,1,0};
10 int bfs(int x, int y, int rooms) {
11     queue<pair<int,int>> q;
12     q.push(make_pair(x,y));
13     d[x][y] = rooms;
14     int cnt = 0;
15     while (!q.empty()) {
16         x = q.front().first;
17         y = q.front().second;
18         q.pop();
19         cnt += 1;
20         for (int k=0; k<4; k++) {
21             int nx = x+dx[k];
22             int ny = y+dy[k];
23             if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
24             if (d[nx][ny] != 0) continue;
25             if (a[x][y] & (1<<k)) continue;
26             q.push(make_pair(nx,ny));
27             d[nx][ny] = rooms;
28         }
29     }
30     return cnt;
31 }
32 int main() {
33     cin >> m >> n;
34     for (int i=0; i<n; i++) {
35         for (int j=0; j<m; j++) {
36             cin >> a[i][j];
37         }
38     }
39     int rooms = 0;
40     for (int i=0; i<n; i++) {
41         for (int j=0; j<m; j++) {
42             if (d[i][j] == 0) {
43                 rooms += 1;
44                 room[rooms] = bfs(i, j, rooms);
45             }
46         }
47     }
48     cout << rooms << '\n';
49     int ans = 0;
50     for (int i=1; i<=rooms; i++) {
51         if (ans < room[i]) {
52             ans = room[i];
53         }
54     }
55     cout << ans << '\n';
56     ans = 0;
57     for (int i=0; i<n; i++) {
58         for (int j=0; j<m; j++) {
59             int x = i;
60             int y = j;
61             for (int k=0; k<4; k++) {
62                 int nx = x+dx[k];
63                 int ny = y+dy[k];
64                 if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
65                 if (d[nx][ny] == d[x][y]) continue;
66                 if (a[x][y] & (1<<k)) {
67                     if (ans < room[d[x][y]]+room[d[nx][ny]]) {
68                         ans = room[d[x][y]]+room[d[nx][ny]];
69                     }
70                 }
71             }
72         }
73     }
74     cout << ans << '\n';
75     return 0;
76 }
77
```



방 번호
↓[][]: (i, j)
방 번호
room[i]: 방 i의 크기

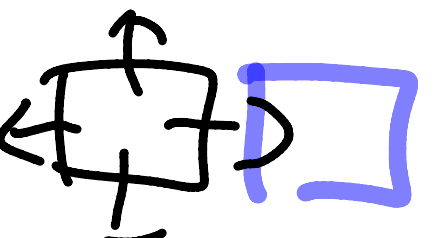
← 방 번호
(x, y) 시작

2^k = (1<<k)

← 방 번호

(NM)

(NM)



O(NM)

C++14

```
1 #include <iostream>
2 #include <map>
3 #include <queue>
4 #include <array>
5 using namespace std;
6 int main() {
7     array<string,3> s;
8     for (int i=0; i<3; i++) {
9         int cnt;
10        cin >> cnt;
11        if (cnt > 0) {
12            cin >> s[i];
13        } else {
14            s[i] = "";
15        }
16    }
17    int cnt[3] = {0, 0, 0};
18    for (int i=0; i<3; i++) {
19        for (int j=0; j<s[i].length(); j++) {
20            cnt[s[i][j]-'A'] += 1;
21        }
22    }
23    map<array<string,3>, int> d;
24    queue<array<string,3>> q;
25    q.push(s);
26    d[s] = 0;
27    while (!q.empty()) {
28        auto now = q.front();
29        q.pop();
30        for (int i=0; i<3; i++) {
31            for (int j=0; j<3; j++) {
32                if (i == j) continue;
33                if (now[i].length() == 0) continue;
34                array<string,3> next(now);
35                next[j].push_back(next[i].back());
36                next[i].pop_back();
37                if (d.count(next) == 0) {
38                    d[next] = d[now] + 1;
39                    q.push(next);
40                }
41            }
42        }
43    }
44    array<string,3> ans;
45    for (int i=0; i<3; i++) {
46        for (int j=0; j<cnt[i]; j++) {
47            ans[i] += (char)('A' + i);
48        }
49    }
50    cout << d[ans] << '\n';
51    return 0;
52 }
```

예: ABC
0 1 2

S[i] = i번 탑의 원판

cnt[i]: i번 원판의 개수

이 부분 코드가 실행되는 과정

↓ " " " " "

6가지

A → B
A → C
B → A
B → C
C → A
C → B

i → j

(next[j].push_back(next[i].back());
next[i].pop_back();
if (d.count(next) == 0) {
d[next] = d[now] + 1;
q.push(next);
}

(B)(C)

결과

메모리

시간

코드 길이

맞았습니다!!

45592 KB

1424 ms

1266 B

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <cstring>
5 using namespace std;
6 int a[100][100];
7 int d[100][100];
8 int dx[] = {0,0,1,-1};
9 int dy[] = {1,-1,0,0};
10 int n, m;
11 vector<pair<int,int>> candi;
12 int ans = -1;
13 void bfs() {
14     memset(d,-1,sizeof(d));
15     queue<pair<int,int>> q;
16     for (int i=0; i<n; i++) {
17         for (int j=0; j<n; j++) {
18             if (a[i][j] == 3) {
19                 q.push(make_pair(i,j));
20                 d[i][j] = 0;
21             }
22         }
23     }
24     while (!q.empty()) {
25         int x, y;
26         tie(x,y) = q.front(); q.pop();
27         for (int k=0; k<4; k++) {
28             int nx = x+dx[k];
29             int ny = y+dy[k];
30             if (0 <= nx && nx < n && 0 <= ny && ny < n) {
31                 if (a[nx][ny] != 1 && d[nx][ny] == -1) {
32                     d[nx][ny] = d[x][y] + 1;
33                     q.push(make_pair(nx,ny));
34                 }
35             }
36         }
37     }
38     int cur = 0;
39     for (int i=0; i<n; i++) {
40         for (int j=0; j<n; j++) {
41             if (a[i][j] != 1) {
42                 if (d[i][j] == -1) return;
43                 if (cur < d[i][j]) cur = d[i][j];
44             }
45         }
46     }
47     if (ans == -1 || ans > cur) {
48         ans = cur;
49     }
50 }
51 void go(int index, int cnt) {
52     if (index == candi.size()) {
53         if (cnt == m) {
54             bfs();
55         }
56     } else {
57         int x, y;
58         tie(x,y) = candi[index];
59         a[x][y] = 3;
60         go(index+1, cnt+1);
61         a[x][y] = 0;
62         go(index+1, cnt);
63     }
64 }
65 int main() {
66     cin >> n >> m;
67     for (int i=0; i<n; i++) {
68         for (int j=0; j<n; j++) {
69             cin >> a[i][j];
70             if (a[i][j] == 2) {
71                 a[i][j] = 0;
72                 candi.push_back(make_pair(i,j));
73             }
74         }
75     }
76     go(0,0);
77     cout << ans << '\n';
78     return 0;
79 }
80
```

결과

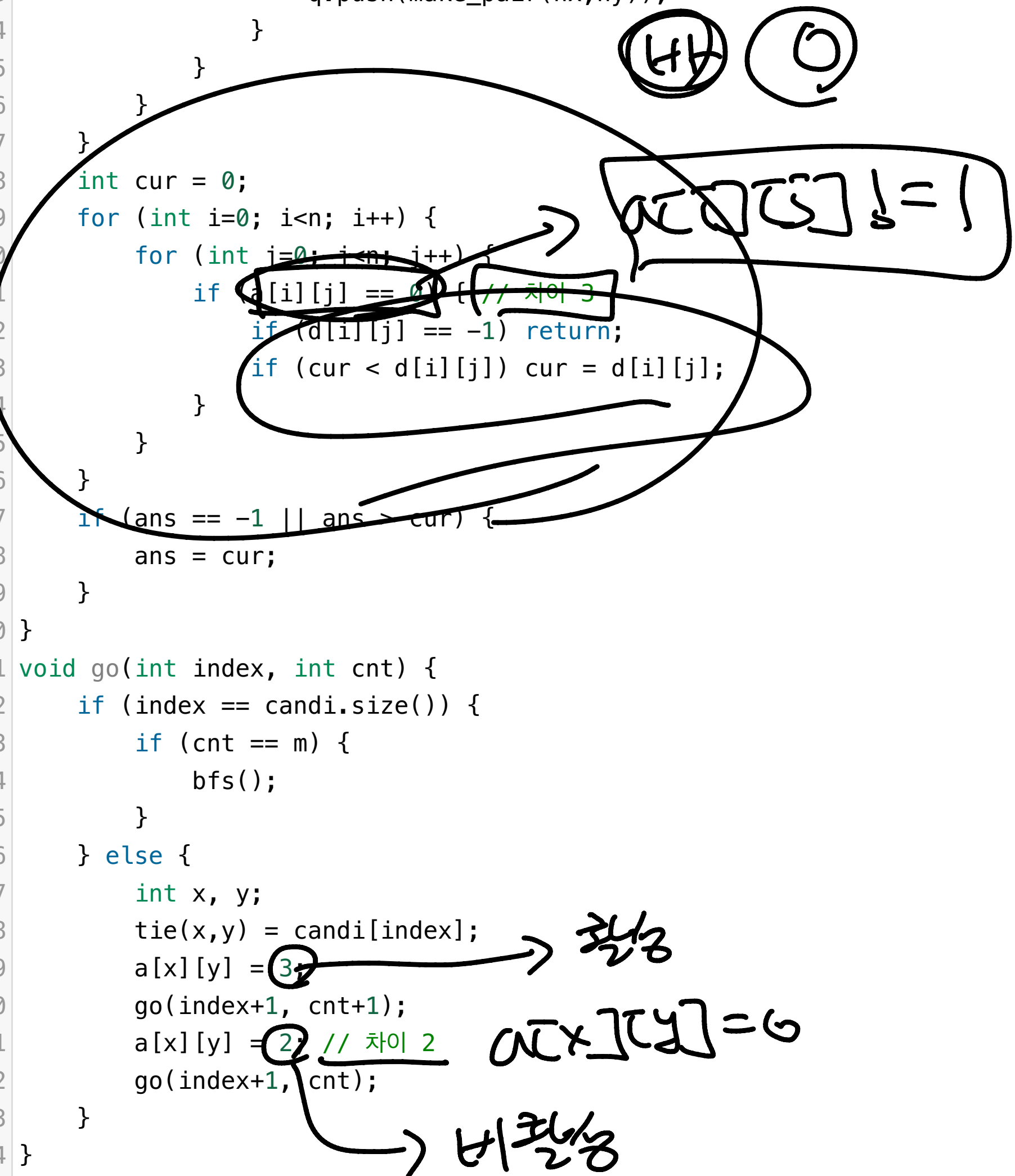
메모리

시간

코드 길이

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <cstring>
5 using namespace std;
6 int a[100][100];
7 int d[100][100];
8 int dx[] = {0,0,1,-1};
9 int dy[] = {1,-1,0,0};
10 int n, m;
11 vector<pair<int,int>> candi;
12 int ans = -1;
13 void bfs() {
14     memset(d,-1,sizeof(d));
15     queue<pair<int,int>> q;
16     for (int i=0; i<n; i++) {
17         for (int j=0; j<n; j++) {
18             if (a[i][j] == 3) {
19                 q.push(make_pair(i,j));
20                 d[i][j] = 0;
21             }
22         }
23     }
24     while (!q.empty()) {
25         int x, y;
26         tie(x,y) = q.front(); q.pop();
27         for (int k=0; k<4; k++) {
28             int nx = x+dx[k];
29             int ny = y+dy[k];
30             if (0 <= nx && nx < n && 0 <= ny && ny < n) {
31                 if (a[nx][ny] != 1 && d[nx][ny] == -1) {
32                     d[nx][ny] = d[x][y] + 1;
33                     q.push(make_pair(nx,ny));
34                 }
35             }
36         }
37     }
38     int cur = 0;
39     for (int i=0; i<n; i++) {
40         for (int j=0; j<n; j++) {
41             if (a[i][j] == 0) { // 차이 3
42                 if (d[i][j] == -1) return;
43                 if (cur < d[i][j]) cur = d[i][j];
44             }
45         }
46     }
47     if (ans == -1 || ans > cur) {
48         ans = cur;
49     }
50 }
51 void go(int index, int cnt) {
52     if (index == candi.size()) {
53         if (cnt == m) {
54             bfs();
55         }
56     } else {
57         int x, y;
58         tie(x,y) = candi[index];
59         a[x][y] = 3;
60         go(index+1, cnt+1);
61         a[x][y] = 2; // 차이 2
62         go(index+1, cnt);
63     }
64 }
65 int main() {
66     cin >> n >> m;
67     for (int i=0; i<n; i++) {
68         for (int j=0; j<n; j++) {
69             cin >> a[i][j];
70             if (a[i][j] == 2) {
71                 // 차이 1 (연구소 2는 a[i][j] = 0; 이 적혀있음)
72                 candi.push_back(make_pair(i,j));
73             }
74         }
75     }
76     go(0,0);
77     cout << ans << '\n';
78     return 0;
79 }
80
```



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.