

BFS (연결)

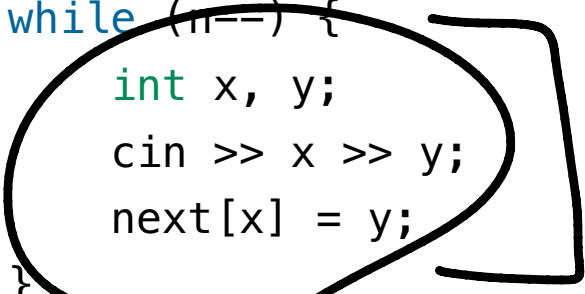
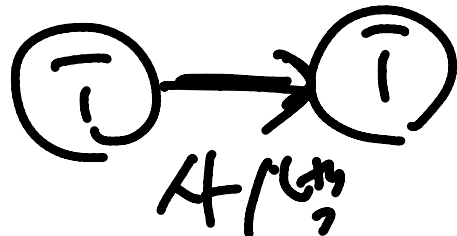
소스코드

최백준 choi@startlink.io

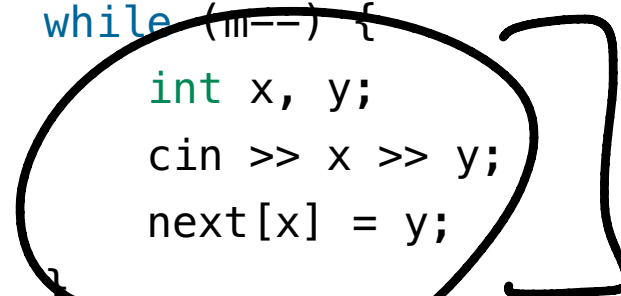
C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <queue>
4 #define next _next
5 using namespace std;
6 int dist[101];
7 int next[101];
8 int main() {
9     int n, m;
10    cin >> n >> m;
11    for (int i=1; i<=100; i++) {
12        next[i] = i;
13        dist[i] = -1;
14    }
15    while (n-->0) {
16        int x, y;
17        cin >> x >> y;
18        next[x] = y;
19    }
20    while (m-->0) {
21        int x, y;
22        cin >> x >> y;
23        next[x] = y;
24    }
25    dist[1] = 0;
26    queue<int> q;
27    q.push(1);
28    while (!q.empty()) {
29        int x = q.front(); q.pop();
30        for (int i=1; i<=6; i++) {
31            int y = x+i;
32            if (y > 100) continue;
33            y = next[y];
34            if (dist[y] == -1) {
35                dist[y] = dist[x] + 1;
36                q.push(y);
37            }
38        }
39    }
40    cout << dist[100] << '\n';
41    return 0;
42 }
43
```

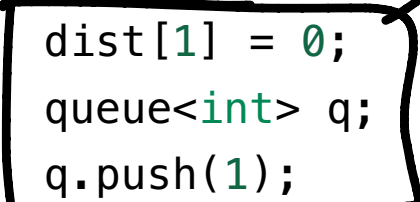
1 ~ 100



12

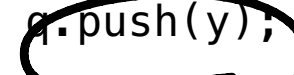
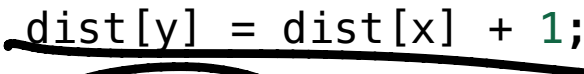
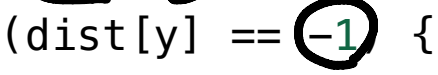


13



14

pop



BFS

결과

메모리

시간

코드 길이

맞았습니다!!

1988 KB

0 ms

835 B

C++14

```

1 #include <iostream>
2 #include <tuple>
3 #include <cstring>
4 #include <queue>
5 using namespace std;
6 int dx[] = {-2,-2,0,0,2,2};
7 int dy[] = {-1,1,-2,2,-1,1};
8 int dist[200][200];
9 int main() {
10     int n;
11     cin >> n;
12     int sx,sy,ex,ey;
13     cin >> sx >> sy >> ex >> ey;
14     memset(dist,-1,sizeof(dist));
15     dist[sx][sy] = 0;
16     queue<pair<int,int>> q;
17     q.push(make_pair(sx,sy));
18     while (!q.empty()) {
19         int x, y;
20         tie(x, y) = q.front(); q.pop();
21         for (int k=0; k<6; k++) {
22             int nx = x+dx[k];
23             int ny = y+dy[k];
24             if (0 <= nx && nx < n && 0 <= ny && ny < n) {
25                 if (dist[nx][ny] == -1) {
26                     q.push(make_pair(nx,ny));
27                     dist[nx][ny] = dist[x][y] + 1;
28                 }
29             }
30         }
31     }
32     cout << dist[ex][ey] << '\n';
33     return 0;
34 }
35
36

```

X행 Y열
 (x,y)
 200^2
 -1
 push
 방문
 방문
 방문
 방문

결과

메모리

시간

코드 길이

맞았습니다!!

2144 KB

0 ms

850 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 int n, m;
5 int a[10][10];
6 int b[10][10];
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 int bfs() {
10     queue<pair<int,int>> q;
11     for (int i=0; i<n; i++) {
12         for (int j=0; j<m; j++) {
13             b[i][j] = a[i][j];
14             if (b[i][j] == 2) {
15                 q.push(make_pair(i,j));
16             }
17         }
18     }
19     while (!q.empty()) {
20         int x = q.front().first;
21         int y = q.front().second;
22         q.pop();
23         for (int k=0; k<4; k++) {
24             int nx = x+dx[k];
25             int ny = y+dy[k];
26             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
27                 if (b[nx][ny] == 0) {
28                     b[nx][ny] = 2;
29                     q.push(make_pair(nx,ny));
30                 }
31             }
32         }
33     }
34     int cnt = 0;
35     for (int i=0; i<n; i++) {
36         for (int j=0; j<m; j++) {
37             if (b[i][j] == 0) {
38                 cnt += 1;
39             }
40         }
41     }
42     return cnt;
43 }
44 int main() {
45     cin >> n >> m;
46     for (int i=0; i<n; i++) {
47         for (int j=0; j<m; j++) {
48             cin >> a[i][j];
49         }
50     }
51     int ans = 0;
52     for (int x1=0; x1<n; x1++) {
53         for (int y1=0; y1<m; y1++) {
54             if (a[x1][y1] != 0) continue;
55             for (int x2=0; x2<n; x2++) {
56                 for (int y2=0; y2<m; y2++) {
57                     if (a[x2][y2] != 0) continue;
58                     for (int x3=0; x3<n; x3++) {
59                         for (int y3=0; y3<m; y3++) {
60                             if (a[x3][y3] != 0) continue;
61                             if (x1 == x2 && y1 == y2) continue;
62                             if (x1 == x3 && y1 == y3) continue;
63                             if (x2 == x3 && y2 == y3) continue;
64                             a[x1][y1] = 1;
65                             a[x2][y2] = 1;
66                             a[x3][y3] = 1;
67                             int cur = bfs();
68                             if (ans < cur) ans = cur;
69                             a[x1][y1] = 0;
70                             a[x2][y2] = 0;
71                             a[x3][y3] = 0;
72                         }
73                     }
74                 }
75             }
76         }
77     }
78     cout << ans << '\n';
79     return 0;
80 }
```

C++14

```
1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 int n, m;
5 int a[10][10];
6 int b[10][10];
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 void dfs(int x, int y) {
10     for (int k=0; k<4; k++) {
11         int nx = x+dx[k];
12         int ny = y+dy[k];
13         if (0 <= nx && nx < n && 0 <= ny && ny < m) {
14             if (b[nx][ny] == 0) {
15                 b[nx][ny] = 2;
16                 dfs(nx,ny);
17             }
18         }
19     }
20 }
21 int dfs() {
22     for (int i=0; i<n; i++) {
23         for (int j=0; j<m; j++) {
24             b[i][j] = a[i][j];
25         }
26     }
27     for (int i=0; i<n; i++) {
28         for (int j=0; j<m; j++) {
29             if (b[i][j] == 2) {
30                 dfs(i, j);
31             }
32         }
33     }
34     int cnt = 0;
35     for (int i=0; i<n; i++) {
36         for (int j=0; j<m; j++) {
37             if (b[i][j] == 0) {
38                 cnt += 1;
39             }
40         }
41     }
42     return cnt;
43 }
44 int main() {
45     cin >> n >> m;
46     for (int i=0; i<n; i++) {
47         for (int j=0; j<m; j++) {
48             cin >> a[i][j];
49         }
50     }
51     int ans = 0;
52     for (int x1=0; x1<n; x1++) {
53         for (int y1=0; y1<m; y1++) {
54             if (a[x1][y1] != 0) continue;
55             for (int x2=0; x2<n; x2++) {
56                 for (int y2=0; y2<m; y2++) {
57                     if (a[x2][y2] != 0) continue;
58                     for (int x3=0; x3<n; x3++) {
59                         for (int y3=0; y3<m; y3++) {
60                             if (a[x3][y3] != 0) continue;
61                             if (x1 == x2 && y1 == y2) continue;
62                             if (x1 == x3 && y1 == y3) continue;
63                             if (x2 == x3 && y2 == y3) continue;
64                             a[x1][y1] = 1;
65                             a[x2][y2] = 1;
66                             a[x3][y3] = 1;
67                             int cur = dfs();
68                             if (ans < cur) ans = cur;
69                             a[x1][y1] = 0;
70                             a[x2][y2] = 0;
71                             a[x3][y3] = 0;
72                         }
73                     }
74                 }
75             }
76         }
77     }
78     cout << ans << '\n';
79     return 0;
80 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

1988 KB

256 ms

2143 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 using namespace std;
4 bool check[1501][1501];
5 int sum;
6 void go(int x, int y) {
7     if (check[x][y]) return;
8     check[x][y] = true;
9     int a[3] = {x, y, sum-x-y};
10    for (int i=0; i<3; i++) {
11        for (int j=0; j<3; j++) {
12            if (a[i] < a[j]) {
13                int b[3] = {x, y, sum-x-y};
14                b[i] += a[i];
15                b[j] -= a[i];
16                go(b[0], b[1]);
17            }
18        }
19    }
20 }
21 int main() {
22     int x, y, z;
23     cin >> x >> y >> z;
24     sum = x + y + z;
25     if (sum % 3 != 0) {
26         cout << 0 << '\n';
27         return 0;
28     }
29     go(x, y);
30     if (check[sum/3][sum/3]) {
31         cout << 1 << '\n';
32     } else {
33         cout << 0 << '\n';
34     }
35     return 0;
36 }
37
```

결과	메모리	시간	코드 길이
----	-----	----	-------

맞았습니다!!	32688 KB	52 ms	747 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 }
```

가는
[0, 1]

(0, 0), 0

벽을 부수는 횟수
z

(x, y) → (nx, ny)

벽
부
수

if (a[nx][ny] == 0 && d[nx][ny][z] == 0) {
d[nx][ny][z] = d[x][y][z] + 1;
q.push(make_tuple(nx, ny, z));
}
if (z == 0 && a[nx][ny] == 1 && d[nx][ny][z+1] == 0) {
d[nx][ny][z+1] = d[x][y][z] + 1;
q.push(make_tuple(nx, ny, z+1));
}

결과

메모리

시간

코드 길이

맞았습니다!!

13708 KB

108 ms

1358 B

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <queue>
5 #include <set>
6 #include <tuple>
7 using namespace std;
8 int n, m;
9 int a[1000][1000];
10 bool check[1000][1000];
11 int group[1000][1000];
12 vector<int> group_size;
13 int dx[] = {0,0,1,-1};
14 int dy[] = {1,-1,0,0};
15 void bfs(int sx, int sy) {
16     int g = group_size.size();
17     queue<pair<int,int>> q;
18     q.push(make_pair(sx,sy));
19     check[sx][sy] = true;
20     group[sx][sy] = g;
21     int cnt = 1;
22     while (!q.empty()) {
23         int x, y;
24         tie(x, y) = q.front(); q.pop();
25         for (int k=0; k<4; k++) {
26             int nx = x+dx[k];
27             int ny = y+dy[k];
28             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
29                 if (a[nx][ny] == 0 && check[nx][ny] == false) {
30                     q.push(make_pair(nx,ny));
31                     check[nx][ny] = true;
32                     group[nx][ny] = g;
33                     cnt += 1;
34                 }
35             }
36         }
37     }
38     group_size.push_back(cnt);
39 }
40 int main() {
41     cin >> n >> m;
42     for (int i=0; i<n; i++) {
43         string s;
44         cin >> s;
45         for (int j=0; j<m; j++) {
46             a[i][j] = s[j] - '0';
47             check[i][j] = false;
48             group[i][j] = -1;
49         }
50     }
51     for (int i=0; i<n; i++) {
52         for (int j=0; j<m; j++) {
53             if (a[i][j] == 0 && check[i][j] == false) {
54                 bfs(i, j);
55             }
56         }
57     }
58     for (int i=0; i<n; i++) {
59         for (int j=0; j<m; j++) {
60             if (a[i][j] == 0) {
61                 cout << 0;
62             } else {
63                 set<int> near;
64                 for (int k=0; k<4; k++) {
65                     int x = i+dx[k];
66                     int y = j+dy[k];
67                     if (0 <= x && x < n && 0 <= y && y < m) {
68                         if (a[x][y] == 0) {
69                             near.insert(group[x][y]);
70                         }
71                     }
72                 }
73                 int ans = 1;
74                 for (int g : near) {
75                     ans += group_size[g];
76                 }
77                 cout << ans%10;
78             }
79         }
80         cout << '\n';
81     }
82     return 0;
83 }
84
```

결과

메모리

시간

코드 길이

맞았습니다!!

11176 KB

160 ms

2156 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <cstdio>
4 #include <tuple>
5 #include <cstring>
6 using namespace std;
7 int a[1000][1000];
8 int d[1000][1000][11];
9 int dx[] = {0, 0, 1, -1};
10 int dy[] = {1, -1, 0, 0};
11 int main() {
12     int n, m, l;
13     scanf("%d %d %d", &n, &m, &l);
14     for (int i=0; i<n; i++) {
15         for (int j=0; j<m; j++) {
16             scanf("%1d", &a[i][j]);
17         }
18     }
19     queue<tuple<int, int, int>> q;
20     d[0][0][0] = 1;
21     q.push(make_tuple(0, 0, 0));
22     while (!q.empty()) {
23         int x, y, z;
24         tie(x, y, z) = q.front(); q.pop();
25         for (int k=0; k<4; k++) {
26             int nx = x+dx[k];
27             int ny = y+dy[k];
28             if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
29             if (a[nx][ny] == 0 && d[nx][ny][z] == 0) {
30                 d[nx][ny][z] = d[x][y][z] + 1;
31                 q.push(make_tuple(nx, ny, z));
32             }
33             if (z+1 <= l && a[nx][ny] == 1 && d[nx][ny][z+1] == 0) {
34                 d[nx][ny][z+1] = d[x][y][z] + 1;
35                 q.push(make_tuple(nx, ny, z+1));
36             }
37         }
38     }
39     int ans = -1;
40     for (int i=0; i<=l; i++) {
41         if (d[n-1][m-1][i] == 0) continue;
42         if (ans == -1) {
43             ans = d[n-1][m-1][i];
44         } else if (ans > d[n-1][m-1][i]) {
45             ans = d[n-1][m-1][i];
46         }
47     }
48     cout << ans << '\n';
49     return 0;
50 }
51
```

결과

메모리

시간

코드 길이

맞았습니다!!

48996 KB

332 ms

1356 B

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <cstdio>
4 #include <tuple>
5 #include <cstring>
6 using namespace std;
7 int a[1000][1000];
8 int d[1000][1000][11][2];
9 int dx[] = {0, 0, 1, -1};
10 int dy[] = {1, -1, 0, 0};
11 int main() {
12     int n, m, l;
13     scanf("%d %d %d",&n,&m, &l);
14     for (int i=0; i<n; i++) {
15         for (int j=0; j<m; j++) {
16             scanf("%1d",&a[i][j]);
17         }
18     }
19     queue<tuple<int,int,int,int>> q;
20     d[0][0][0][0] = 1;
21     q.push(make_tuple(0,0,0,0));
22     while (!q.empty()) {
23         int x, y, z, night;
24         tie(x,y,z, night) = q.front(); q.pop();
25         for (int k=0; k<4; k++) {
26             int nx = x+dx[k];
27             int ny = y+dy[k];
28             if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
29             if (a[nx][ny] == 0 && d[nx][ny][z][1-night] == 0) {
30                 d[nx][ny][z][1-night] = d[x][y][z][night] + 1;
31                 q.push(make_tuple(nx,ny,z,1-night));
32             }
33             if (night == 0 && z+1 <= l && a[nx][ny] == 1 && d[nx][ny][z+1][1-night] == 0) {
34                 d[nx][ny][z+1][1-night] = d[x][y][z][night] + 1;
35                 q.push(make_tuple(nx,ny,z+1,1-night));
36             }
37         }
38         if (d[x][y][z][1-night] == 0) {
39             d[x][y][z][1-night] = d[x][y][z][night] + 1;
40             q.push(make_tuple(x,y,z,1-night));
41         }
42     }
43     int ans = -1;
44     for (int j=0; j<2; j++) {
45         for (int i=0; i<=l; i++) {
46             if (d[n-1][m-1][i][j] == 0) continue;
47             if (ans == -1) {
48                 ans = d[n-1][m-1][i][j];
49             } else if (ans > d[n-1][m-1][i][j]) {
50                 ans = d[n-1][m-1][i][j];
51             }
52         }
53     }
54     cout << ans << '\n';
55     return 0;
56 }
```

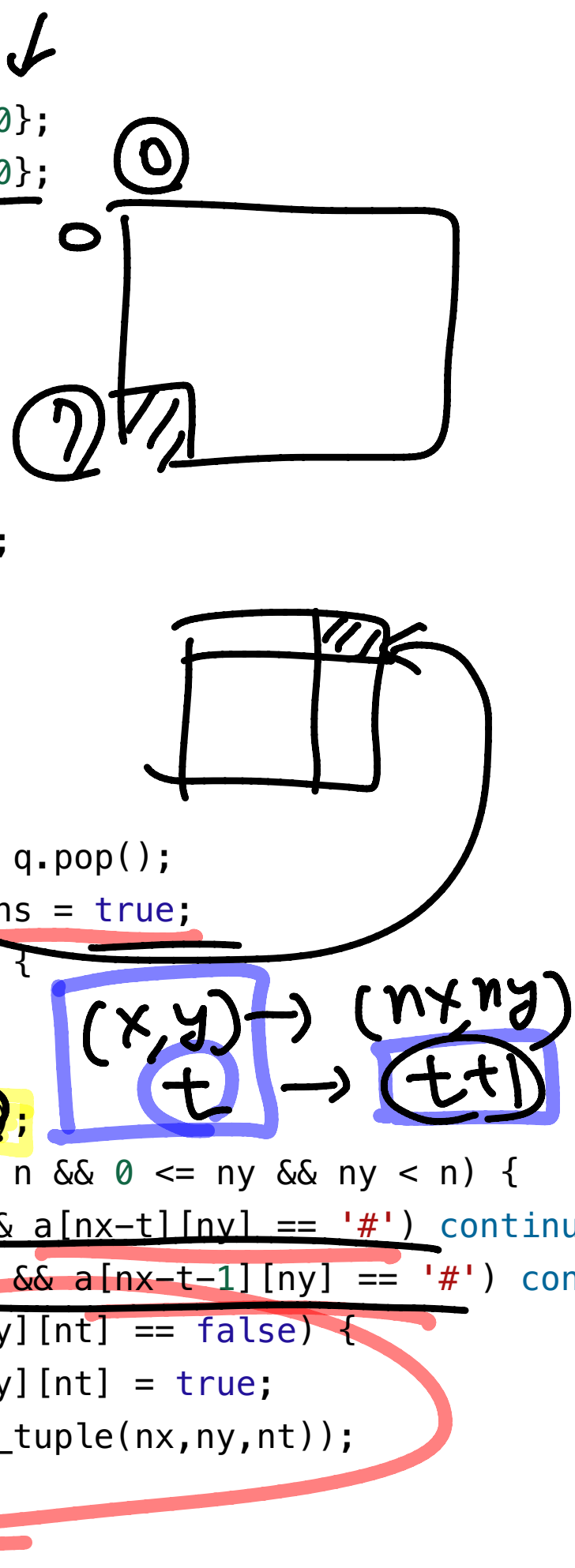
관, ω, (벽)

0, 1

night → 1-night
0 → 1
1 → 0

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <vector>
4 #include <string>
5 #include <queue>
6 using namespace std;
7 bool check[8][8][9];
8 int dx[] = {0,0,1,-1,1,-1,1,-1};
9 int dy[] = {1,-1,0,0,1,1,-1,-1};
10 int main() {
11     int n = 8;
12     vector<string> a(n);
13     for (int i=0; i<n; i++) {
14         cin >> a[i];
15     }
16     queue<tuple<int,int,int>> q;
17     check[7][0][0] = true;
18     q.push(make_tuple(7,0,0));
19     bool ans = false;
20     while (!q.empty()) {
21         int x, y, t;
22         tie(x,y,t) = q.front(); q.pop();
23         if (x == 0 && y == 7) ans = true;
24         for (int k=0; k<9; k++) {
25             int nx = x+dx[k];
26             int ny = y+dy[k];
27             int nt = min(t+1, 8);
28             if (0 <= nx && nx < n && 0 <= ny && ny < n) {
29                 if (nx-t >= 0 && a[nx-t][ny] == '#') continue;
30                 if (nx-t-1 >= 0 && a[nx-t-1][ny] == '#') continue;
31                 if (check[nx][ny][nt] == false) {
32                     check[nx][ny][nt] = true;
33                     q.push(make_tuple(nx,ny,nt));
34                 }
35             }
36         }
37     }
38     cout << (ans ? 1 : 0) << '\n';
39     return 0;
40 }
41
```



남지 →
t-1
t+1

결과	메모리	시간	코드 길이
맞았습니다!!	1992 KB	0 ms	1115 B

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <queue>
4 #include <tuple>
5 #include <vector>
6 using namespace std;
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 tuple<int,int,int> bfs(vector<vector<int>> &a, int x, int y, int size) {
10     int n = a.size();
11     vector<tuple<int,int,int>> ans;
12     vector<vector<int>> d(n, vector<int> (n, -1));
13     queue<pair<int,int>> q;
14     q.push(make_pair(x,y));
15     d[x][y] = 0;
16     while (!q.empty()) {
17         tie(x,y) = q.front();
18         q.pop();
19         for (int k=0; k<4; k++) {
20             int nx = x+dx[k];
21             int ny = y+dy[k];
22             if (0 <= nx && nx < n && 0 <= ny && ny < n && d[nx][ny] == -1) {
23                 bool ok = false;
24                 bool eat = false;
25                 // 아기 상어는 자신의 크기보다 큰 물고기가 있는 칸은 지나갈 수 없고, 나머지 칸은 모두 지나갈 수 있다.
26                 if (a[nx][ny] == 0) {
27                     ok = true;
28                 } else if (a[nx][ny] < size) { // 아기 상어는 자신의 크기보다 작은 물고기만 먹을 수 있다.
29                     ok = eat = true;
30                 } else if (a[nx][ny] == size) { // 크기가 같은 물고기는 먹을 수 없지만, 그 물고기가 있는 칸은 지나갈 수 있다.
31                     ok = true;
32                 }
33                 if (ok) {
34                     q.push(make_pair(nx,ny));
35                     d[nx][ny] = d[x][y] + 1;
36                     if (eat) {
37                         ans.push_back(make_tuple(d[nx][ny], nx, ny));
38                     }
39                 }
40             }
41         }
42     }
43     if (ans.empty()) {
44         return make_tuple(-1,-1,-1);
45     }
46     sort(ans.begin(), ans.end());
47     return ans[0];
48 }
49 int main() {
50     int n;
51     cin >> n;
52     vector<vector<int>> a(n, vector<int>(n, 0));
53     int x, y;
54     for (int i=0; i<n; i++) {
55         for (int j=0; j<n; j++) {
56             cin >> a[i][j];
57             if (a[i][j] == 9) {
58                 tie(x, y) = make_pair(i, j);
59                 a[i][j] = 0;
60             }
61         }
62     }
63     int ans = 0;
64     int size = 2;
65     int exp = 0;
66     while (true) {
67         int nx, ny, dist;
68         tie(dist, nx, ny) = bfs(a, x, y, size);
69         if (dist == -1) break;
70         a[nx][ny] = 0;
71         ans += dist;
72         exp += 1;
73         if (size == exp) {
74             size += 1;
75             exp = 0;
76         }
77         tie(x,y) = make_pair(nx,ny);
78     }
79     cout << ans << '\n';
80     return 0;
81 }
82
```

결과

메모리

시간

코드 길이

맞았습니다!!

1992 KB

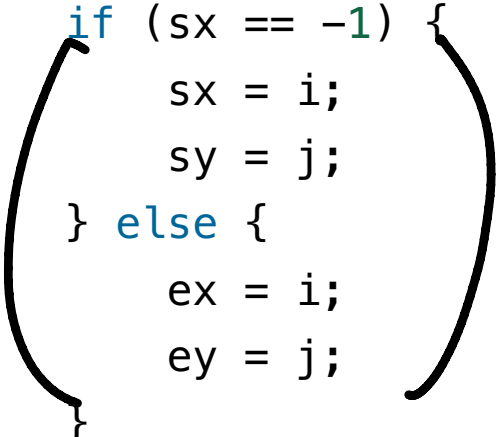
4 ms

2461 B

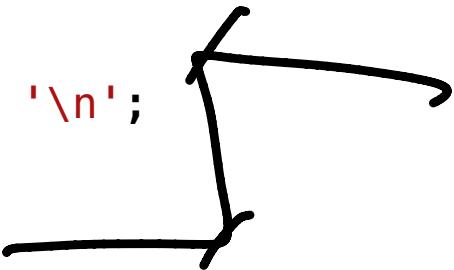
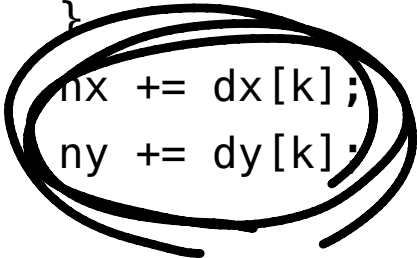
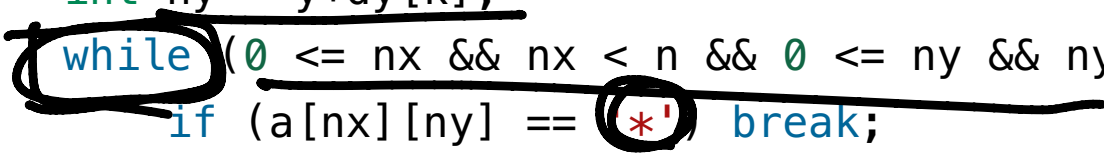
C++14

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

4방향



k번씩 방향



C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <queue>
4 #include <string>
5 #include <algorithm>
6 using namespace std;
7 bool prime[10001];
8 bool c[10001];
9 int d[10001];
10 int change(int num, int index, int digit) {
11     if (index == 0 && digit == 0) return -1;
12     string s = to_string(num);
13     s[index] = digit + '0';
14     return stoi(s);
15 }
16 int main() {
17     for (int i=2; i<=10000; i++) {
18         if (prime[i] == false) {
19             for (int j=i*i; j <= 10000; j+=i) {
20                 prime[j] = true;
21             }
22         }
23     }
24     for (int i=0; i<=10000; i++) {
25         prime[i] = !prime[i];
26     }
27     int t;
28     cin >> t;
29     while (t--) {
30         int n, m;
31         cin >> n >> m;
32         memset(c, false, sizeof(c));
33         memset(d, 0, sizeof(d));
34         d[n] = 0;
35         c[n] = true;
36         queue<int> q;
37         q.push(n);
38         while (!q.empty()) {
39             int now = q.front();
40             q.pop();
41             for (int i=0; i<4; i++) {
42                 for (int j=0; j<=9; j++) {
43                     int next = change(now, i, j);
44                     if (next != -1) {
45                         if (prime[next] && c[next] == false) {
46                             q.push(next);
47                             d[next] = d[now] + 1;
48                             c[next] = true;
49                         }
50                     }
51                 }
52             }
53         }
54         cout << d[m] << '\n';
55     }
56     return 0;
57 }
```

num

Index 12213

digit

0 1 2 3

n → m

1번씩 자릿수 늘려
33 바이트

10000

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <vector>
4 #include <string>
5 using namespace std;
6 int dx[] = {0,0,1,-1};
7 int dy[] = {1,-1,0,0};
8 bool can(bool blind, char from, char to) {
9     if (from == to) return true;
10    if (blind) {
11        if (from == 'R' && to == 'G') return true;
12        if (from == 'G' && to == 'R') return true;
13    }
14    return false;
15 }
16 int go(vector<string> &a, bool blind = false) {
17     int n = a.size();
18     vector<vector<bool>> check(n, vector<bool>(n, false));
19     int ans = 0;
20     for (int i=0; i<n; i++) {
21         for (int j=0; j<n; j++) {
22             if (check[i][j] == false) {
23                 ans += 1;
24                 queue<pair<int,int>> q;
25                 q.push(make_pair(i,j));
26                 check[i][j] = true;
27                 while (!q.empty()) {
28                     int x = q.front().first;
29                     int y = q.front().second;
30                     q.pop();
31                     for (int k=0; k<4; k++) {
32                         int nx = x+dx[k];
33                         int ny = y+dy[k];
34                         if (0 <= nx && nx < n && 0 <= ny && ny < n) {
35                             if (check[nx][ny]) continue;
36                             if (can(blind, a[x][y], a[nx][ny])) {
37                                 check[nx][ny] = true;
38                                 q.push(make_pair(nx,ny));
39                             }
40                         }
41                     }
42                 }
43             }
44         }
45     }
46     return ans;
47 }
48 int main() {
49     int n;
50     cin >> n;
51     vector<string> a(n);
52     for (int i=0; i<n; i++) {
53         cin >> a[i];
54     }
55     cout << go(a) << ' ' << go(a, true) << '\n';
56     return 0;
57 }
58
```

결과

메모리

시간

코드 길이

맞았습니다!!

1992 KB

0 ms

1670 B

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <string>
5 #include <set>
6 using namespace std;
7 const long long limit = 1000000000LL;
8 int main() {
9     long long s, t;
10    cin >> s >> t;
11    set<long long> check;
12    queue<pair<long long, string>> q;
13    q.push(make_pair(s, ""));
14    check.insert(s);
15    while (!q.empty()) {
16        long long x;
17        string str;
18        tie(x, str) = q.front(); q.pop();
19        if (x == t) {
20            if (str.length() == 0) {
21                str = "0";
22            }
23            cout << str << '\n';
24            return 0;
25        }
26        if (0 <= x*x && x*x <= limit && check.count(x*x) == 0) {
27            q.push(make_pair(x*x, str+"*"));
28            check.insert(x*x);
29        }
30        if (0 <= x+x && x+x <= limit && check.count(x+x) == 0) {
31            q.push(make_pair(x+x, str+"+"));
32            check.insert(x+x);
33        }
34        if (0 <= x-x && x-x <= limit && check.count(x-x) == 0) {
35            q.push(make_pair(x-x, str+"-"));
36            check.insert(x-x);
37        }
38        if (x != 0 && 0 <= x/x && x/x <= limit && check.count(x/x) == 0) {
39            q.push(make_pair(x/x, str+"/"));
40            check.insert(x/x);
41        }
42    }
43    cout << -1 << '\n';
44    return 0;
45 }
```

log

DSUR

결과

메모리

시간

코드 길이

맞았습니다!!

1992 KB

0 ms

1244 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.