

BFS 2

소스코드

최백준 [choi@startlink.io](mailto:choi@startlink.io)

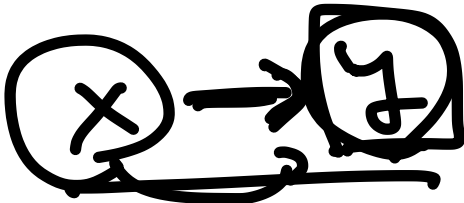
---

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <vector>
4 #include <string>
5 #include <queue>
6 using namespace std;
7 int main() {
8     int t;
9     cin >> t;
10    while (t--) {
11        int n;
12        cin >> n;
13        vector<int> from(n,-1);
14        vector<int> how(n,-1);
15        vector<int> dist(n,-1);
16        queue<int> q;
17        q.push(1%n);
18        dist[1%n] = 0;
19        how[1%n] = 1;
20        while (!q.empty()) {
21            int now = q.front(); q.pop();
22            for (int i=0; i<=1; i++) {
23                int next = (now*10+i)%n;
24                if (dist[next] == -1) {
25                    dist[next] = dist[now] + 1;
26                    from[next] = now;
27                    how[next] = i;
28                    q.push(next);
29                }
30            }
31        }
32        if (dist[0] == -1) {
33            cout << "BRAK" << '\n';
34        } else {
35            string ans = "";
36            for (int i=0; i!=-1; i=from[i]) {
37                ans += to_string(how[i]);
38            }
39            reverse(ans.begin(),ans.end());
40            cout << ans << '\n';
41        }
42    }
43    return 0;
44 }
```

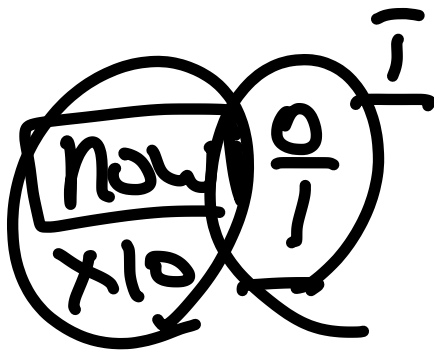
$1 \leq N$   
0

BFS

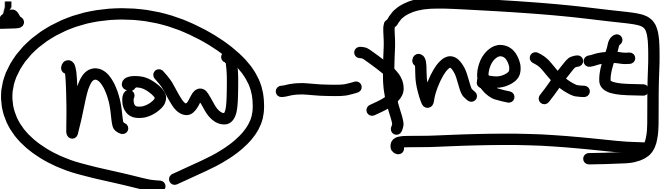


$fun[y] = x$

뒤의 값 + 1



값 + 1



우리가 0 = N의 배수

우리가 0

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <queue>
4 using namespace std;
5 int dist[500001];
6 int main() {
7     int n, k;
8     cin >> n >> k;
9     if (n == k) {
10         cout << 0 << '\n';
11         return 0;
12     }
13     memset(dist, -1, sizeof(dist));
14     queue<int> q;
15     q.push(n);
16     dist[n] = 0;
17     for (int t=1;; t++) {
18         k += t;
19         if (k > 500000) break;
20         queue<int> nq;
21         while (!q.empty()) {
22             int x = q.front();
23             q.pop();
24             for (int y : {x+1, x-1, 2*x}) {
25                 if (0 <= y && y <= 500000) {
26                     if (dist[y] != t) {
27                         nq.push(y);
28                         dist[y] = t;
29                     }
30                 }
31             }
32         }
33         if (dist[k] == t) {
34             cout << t << '\n';
35             return 0;
36         }
37         q = nq;
38     }
39     cout << -1 << '\n';
40 }
41
```

결과

메모리

시간

코드 길이

맞았습니다!!

7756 KB

4960 ms

876 B

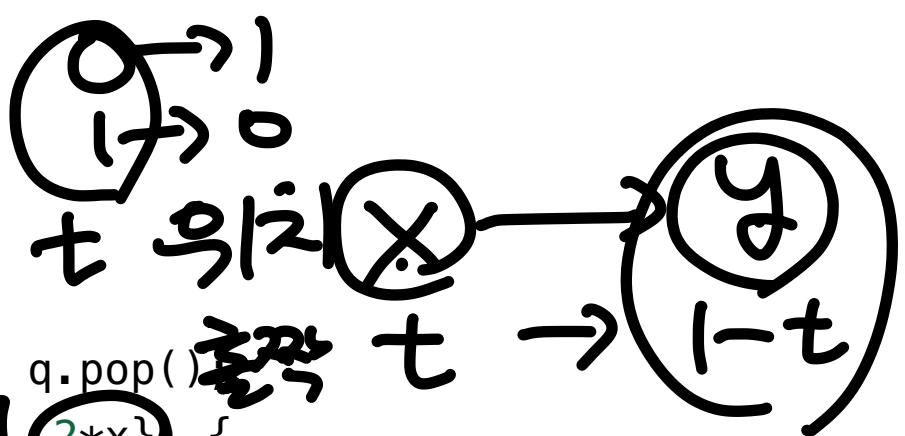
5초

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <cstring>
4 #include <queue>
5 using namespace std;
6 int dist[500001][2];
7 int main() {
8     int n, k;
9     cin >> n >> k;
10    memset(dist, -1, sizeof(dist));
11    queue<pair<int, int>> q;
12    q.push(make_pair(n, 0));
13    dist[n][0] = 0;
14    while (!q.empty()) {
15        int x, t;
16        tie(x, t) = q.front(); q.pop();
17        for (int y : {x+1, x-1, 2*x}) {
18            if (0 <= y && y <= 500000) {
19                if (dist[y][1-t] == -1) {
20                    dist[y][1-t] = dist[x][t] + 1;
21                    q.push(make_pair(y, 1-t));
22                }
23            }
24        }
25    }
26    int ans = -1;
27    for (int t=0;; t++) {
28        k += t;
29        if (k > 500000) break;
30        if (dist[k][t%2] != -1) {
31            ans = t;
32            break;
33        }
34    }
35    cout << ans << '\n';
36    return 0;
37 }
38
```

0초  
1초

0초



바른 42

100만

t초씩 동생의 위치

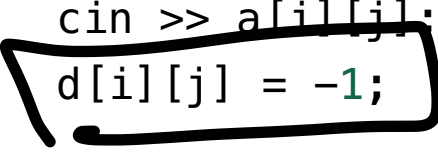
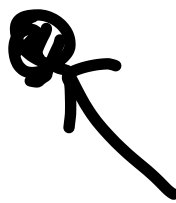
가장 빠른 42

0\_016

결과	메모리	시간	코드 길이
맞았습니다!!	7672 KB	16 ms	845 B

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 using namespace std;
5 int a[1001][1001];
6 int d[1001][1001];
7 int s[1001][1001];
8 int dx[] = {1,-1,0,0};
9 int dy[] = {0,0,1,-1};
10 int sum(int x1, int y1, int x2, int y2) {
11     return s[x2][y2] - s[x1-1][y2] - s[x2][y1-1] + s[x1-1][y1-1];
12 }
13 int main() {
14     ios_base::sync_with_stdio(false);
15     cin.tie(nullptr);
16     int n, m;
17     cin >> n >> m;
18     for (int i=1; i<=n; i++) {
19         for (int j=1; j<=m; j++) {
20             cin >> a[i][j];
21             d[i][j] = -1;
22         }
23     }
24     int h, w;
25     cin >> h >> w;
26     int x1, y1, x2, y2;
27     cin >> x1 >> y1 >> x2 >> y2;
28     for (int i=1; i<=n; i++) {
29         for (int j=1; j<=m; j++) {
30             s[i][j] = s[i-1][j] + s[i][j-1] - s[i-1][j-1] + a[i][j];
31         }
32     }
33     queue<pair<int,int>> q;
34     q.push(make_pair(x1,y1));
35     d[x1][y1] = 0;
36     while (!q.empty()) {
37         int x, y;
38         tie(x, y) = q.front(); q.pop();
39         for (int k=0; k<4; k++) {
40             int nx = x+dx[k];
41             int ny = y+dy[k];
42             if (1 <= nx && nx+h-1 <= n && 1 <= ny && ny+w-1 <= m) {
43                 if (sum(nx,ny,nx+h-1,ny+w-1) == 0) {
44                     if (d[nx][ny] == -1) {
45                         q.push(make_pair(nx,ny));
46                         d[nx][ny] = d[x][y] + 1;
47                     }
48                 }
49             }
50         }
51     }
52     cout << d[x2][y2] << '\n';
53     return 0;
54 }
55
```

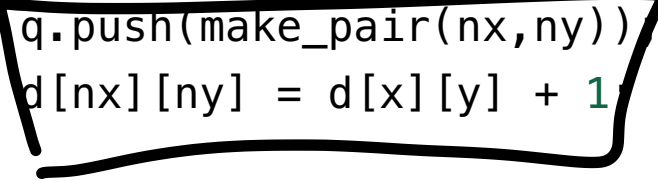


가라!

(x,y) → (nx,ny)



O(NM)



결과

메모리

시간

코드 길이

맞았습니다!!

13860 KB

104 ms

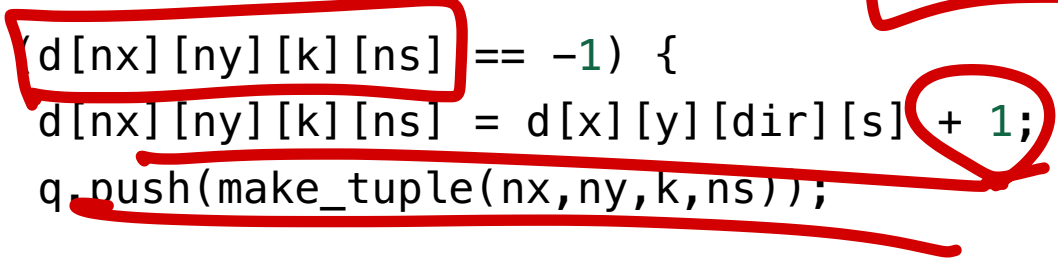
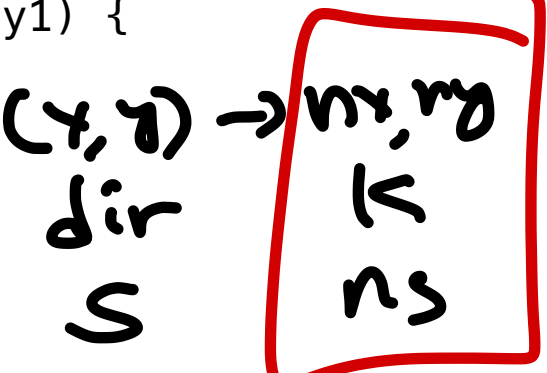
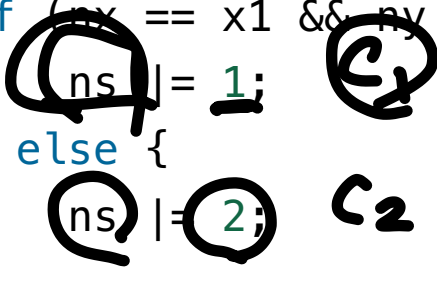
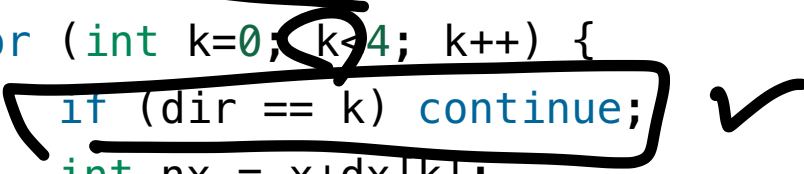
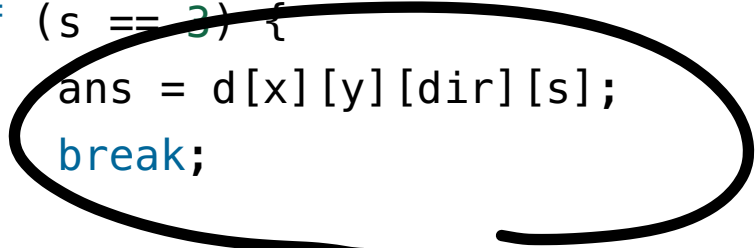
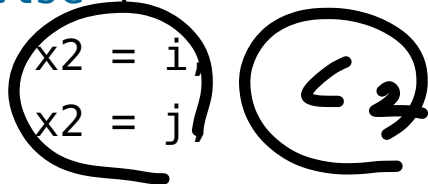
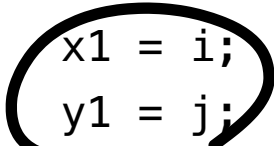
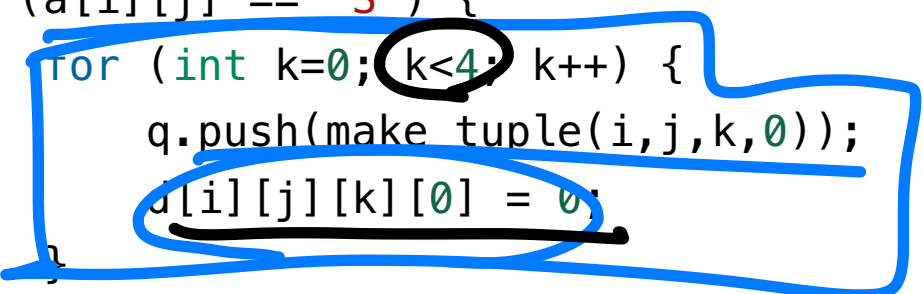
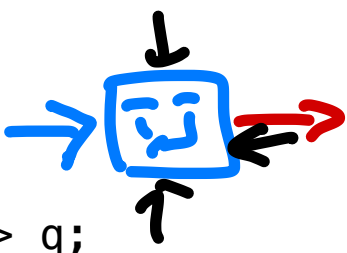
1405 B



C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <cstring>
4 #include <queue>
5 #include <vector>
6 #include <string>
7 using namespace std;
8 int dx[] = {0, 0, 1, -1};
9 int dy[] = {1, -1, 0, 0};
10 int d[51][51][4][4];
11 int main() {
12     int n, m;
13     cin >> n >> m;
14     vector<string> a(n);
15     for (int i=0; i<n; i++) {
16         cin >> a[i];
17     }
18     int ans = -1;
19     int x1,y1,x2,y2;
20     x1 = y1 = x2 = y2 = -1;
21     memset(d,-1,sizeof(d));
22     queue<tuple<int,int,int,int>> q;
23     for (int i=0; i<n; i++) {
24         for (int j=0; j<m; j++) {
25             if (a[i][j] == 'S') {
26                 for (int k=0; k<4; k++) {
27                     q.push(make_tuple(i,j,k,0));
28                     d[i][j][k][0] = 0;
29                 }
30             } else if (a[i][j] == 'C') {
31                 if (x1 == -1) {
32                     x1 = i;
33                     y1 = j;
34                 } else {
35                     x2 = i;
36                     y2 = j;
37                 }
38             }
39         }
40     }
41     while (!q.empty()) {
42         int x,y,dir,s;
43         tie(x,y,dir,s) = q.front();
44         q.pop();
45         if (s == 3) {
46             ans = d[x][y][dir][s];
47             break;
48         }
49         for (int k=0; k<4; k++) {
50             if (dir == k) continue;
51             int nx = x+dx[k];
52             int ny = y+dy[k];
53             if (0 <= nx && nx < n && 0 <= ny && ny < m) {
54                 if (a[nx][ny] != '#') {
55                     int ns = s;
56                     if (a[nx][ny] == 'C') {
57                         if (nx == x1 && ny == y1) {
58                             ns = 1;
59                         } else {
60                             ns = 2;
61                         }
62                     }
63                     if (d[nx][ny][k][ns] == -1) {
64                         d[nx][ny][k][ns] = d[x][y][dir][s] + 1;
65                         q.push(make_tuple(nx,ny,k,ns));
66                     }
67                 }
68             }
69         }
70     }
71     cout << ans << '\n';
72     return 0;
73 }
```

(r,c,k,s)



결과

메모리

시간

코드 길이

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <tuple>
4 #include <queue>
5 #include <vector>
6 using namespace std;
7 int n;
8 int a[10][10];
9 int d[10][10][100][3]; // (x, y, num, piece)
10 queue<tuple<int,int,int,int>> q;
11 int dx1[] = {-2,-1,1,2,2,1,-1,-2};
12 int dy1[] = {1,2,2,1,-1,-2,-2,-1};
13 int dx2[] = {0,0,1,-1};
14 int dy2[] = {1,-1,0,0};
15 int dx3[] = {1,1,-1,-1};
16 int dy3[] = {1,-1,1,-1};
17 int main() {
18     memset(d,-1,sizeof(d));
19     cin >> n;
20     for (int i=0; i<n; i++) {
21         for (int j=0; j<n; j++) {
22             cin >> a[i][j];
23             a[i][j]--;
24             if (a[i][j] == 0) {
25                 for (int k=0; k<3; k++) {
26                     d[i][j][0][k] = 0;
27                     q.push(make_tuple(i,j,0,k));
28                 }
29             }
30         }
31     }
32     int ans = -1;
33     while (!q.empty()) {
34         int x, y, num, piece;
35         tie(x, y, num, piece) = q.front(); q.pop();
36         if (num == n*n-1) {
37             if (ans == -1 || ans > d[x][y][num][piece]) {
38                 ans = d[x][y][num][piece];
39             }
40         }
41         for (int i=0; i<3; i++) {
42             if (piece == i) continue;
43             if (d[x][y][num][i] == -1) {
44                 d[x][y][num][i] = d[x][y][num][piece] + 1;
45                 q.push(make_tuple(x,y,num,i));
46             }
47         }
48         if (piece == 0) { // knight
49             for (int k=0; k<8; k++) {
50                 int nx = x+dx1[k];
51                 int ny = y+dy1[k];
52                 if (0 <= nx && nx < n && 0 <= ny && ny < n) {
53                     int next_num = num;
54                     if (a[nx][ny] == num+1) {
55                         next_num = num+1;
56                     }
57                     if (d[nx][ny][next_num][piece] == -1) {
58                         d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
59                         q.push(make_tuple(nx, ny, next_num, piece));
60                     }
61                 }
62             }
63         } else if (piece == 1) { // rook
64             for (int k=0; k<4; k++) {
65                 for (int l=1; l<=n; l++) {
66                     int nx = x+dx2[k]*l;
67                     int ny = y+dy2[k]*l;
68                     if (0 <= nx && nx < n && 0 <= ny && ny < n) {
69                         int next_num = num;
70                         if (a[nx][ny] == num+1) {
71                             next_num = num+1;
72                         }
73                         if (d[nx][ny][next_num][piece] == -1) {
74                             d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
75                             q.push(make_tuple(nx, ny, next_num, piece));
76                         }
77                     } else {
78                         break;
79                     }
80                 }
81             }
82         } else { // bishop
83             for (int k=0; k<4; k++) {
84                 for (int l=1; l<=n; l++) {
85                     int nx = x+dx3[k]*l;
86                     int ny = y+dy3[k]*l;
87                     if (0 <= nx && nx < n && 0 <= ny && ny < n) {
88                         int next_num = num;
89                         if (a[nx][ny] == num+1) {
90                             next_num = num+1;
91                         }
92                         if (d[nx][ny][next_num][piece] == -1) {
93                             d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
94                             q.push(make_tuple(nx, ny, next_num, piece));
95                         }
96                     } else {
97                         break;
98                     }
99                 }
100             }
101         }
102     }
103     cout << ans << '\n';
104     return 0;
105 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

2104 KB

4 ms

3525 B



C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <tuple>
4 #include <queue>
5 #include <vector>
6 using namespace std;
7 int n;
8 int a[10][10];
9 pair<int,int> d[10][10][100][3]; // (x, y, num, piece)
10 queue<tuple<int,int,int,int>> q;
11 int dx1[] = {-2,-1,1,2,2,1,-1,-2};
12 int dy1[] = {1,2,2,1,-1,-2,-2,-1};
13 int dx2[] = {0,0,1,-1};
14 int dy2[] = {1,-1,0,0};
15 int dx3[] = {1,1,-1,-1};
16 int dy3[] = {1,-1,1,-1};
17 int main() {
18     memset(d,-1,sizeof(d));
19     cin >> n;
20     for (int i=0; i<n; i++) {
21         for (int j=0; j<n; j++) {
22             for (int k=0; k<n*n; k++) {
23                 for (int l=0; l<3; l++) {
24                     d[i][j][k][l] = make_pair(-1,-1);
25                 }
26             }
27         }
28     }
29     for (int i=0; i<n; i++) {
30         for (int j=0; j<n; j++) {
31             cin >> a[i][j];
32             a[i][j] -= 1;
33             if (a[i][j] == 0) {
34                 for (int k=0; k<3; k++) {
35                     d[i][j][0][k] = make_pair(0,0);
36                     q.push(make_tuple(i,j,0,k));
37                 }
38             }
39         }
40     }
41     pair<int,int> ans = make_pair(-1,-1);
42     while (!q.empty()) {
43         int x, y, num, piece;
44         tie(x, y, num, piece) = q.front(); q.pop();
45         auto &p = d[x][y][num][piece];
46         if (num == n*n-1) {
47             if (ans.first == -1 || ans > p) {
48                 ans = p;
49             }
50         }
51         for (int i=0; i<3; i++) {
52             if (piece == i) continue;
53             auto np = make_pair(p.first+1, p.second+1);
54             if (d[x][y][num][i].first == -1 || d[x][y][num][i] > np) {
55                 d[x][y][num][i] = np;
56                 q.push(make_tuple(x,y,num,i));
57             }
58         }
59         if (piece == 0) { // knight
60             for (int k=0; k<8; k++) {
61                 int nx = x+dx1[k];
62                 int ny = y+dy1[k];
63                 if (0 <= nx && nx < n && 0 <= ny && ny < n) {
64                     int next_num = num;
65                     if (a[nx][ny] == num+1) {
66                         next_num = num+1;
67                     }
68                     auto np = make_pair(p.first+1, p.second);
69                     if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
70                         d[nx][ny][next_num][piece] = np;
71                         q.push(make_tuple(nx, ny, next_num, piece));
72                     }
73                 }
74             }
75         } else if (piece == 1) { // rook
76             for (int k=0; k<4; k++) {
77                 for (int l=1;; l++) {
78                     int nx = x+dx2[k]*l;
79                     int ny = y+dy2[k]*l;
80                     if (0 <= nx && nx < n && 0 <= ny && ny < n) {
81                         int next_num = num;
82                         if (a[nx][ny] == num+1) {
83                             next_num = num+1;
84                         }
85                         auto np = make_pair(p.first+1, p.second);
86                         if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
87                             d[nx][ny][next_num][piece] = np;
88                             q.push(make_tuple(nx, ny, next_num, piece));
89                         }
90                     } else {
91                         break;
92                     }
93                 }
94             }
95         } else { // bishop
96             for (int k=0; k<4; k++) {
97                 for (int l=1;; l++) {
98                     int nx = x+dx3[k]*l;
99                     int ny = y+dy3[k]*l;
100                     if (0 <= nx && nx < n && 0 <= ny && ny < n) {
101                         int next_num = num;
102                         if (a[nx][ny] == num+1) {
103                             next_num = num+1;
104                         }
105                         auto np = make_pair(p.first+1, p.second);
106                         if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
107                             d[nx][ny][next_num][piece] = np;
108                             q.push(make_tuple(nx, ny, next_num, piece));
109                         }
110                     } else {
111                         break;
112                     }
113                 }
114             }
115         }
116     }
117     cout << ans.first << ' ' << ans.second << '\n';
118     return 0;
119 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

2224 KB

4 ms

4175 B



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4 #include <string>
5 using namespace std;
6 string a[1500];
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 bool wcheck[1500][1500];
10 bool scheck[1500][1500];
11 int main() {
12     int n, m;
13     cin >> n >> m;
14     int sx,sy,ex,ey;
15     sx=sy=ex=ey=-1;
16     queue<pair<int,int>> nswan, water, nwater;
17     for (int i=0; i<n; i++) {
18         cin >> a[i];
19         for (int j=0; j<m; j++) {
20             if (a[i][j] == 'L') {
21                 if (sx == -1) {
22                     sx = i;
23                     sy = j;
24                 } else {
25                     ex = i;
26                     ey = j;
27                 }
28                 a[i][j] = '.';
29             }
30             if (a[i][j] == '.') {
31                 water.push(make_pair(i,j));
32                 wcheck[i][j] = true;
33             }
34         }
35     }
36     nswan.push(make_pair(sx,sy));
37     scheck[sx][sy] = true;
38     for (int i=0; i<n; i++) {
39         while (!water.empty()) {
40             int x = water.front().first;
41             int y = water.front().second;
42             water.pop();
43             a[x][y] = '.';
44             for (int k=0; k<4; k++) {
45                 int nx = x+dx[k];
46                 int ny = y+dy[k];
47                 if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
48                 if (wcheck[nx][ny]) continue;
49                 if (a[nx][ny] == '.') {
50                     water.push(make_pair(nx,ny));
51                     wcheck[nx][ny] = true;
52                 } else {
53                     nwater.push(make_pair(nx,ny));
54                     wcheck[nx][ny] = true;
55                 }
56             }
57         }
58         while (!nswan.empty()) {
59             int x = nswan.front().first;
60             int y = nswan.front().second;
61             nswan.pop();
62             for (int k=0; k<4; k++) {
63                 int nx = x+dx[k];
64                 int ny = y+dy[k];
65                 if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
66                 if (scheck[nx][ny]) continue;
67                 if (a[nx][ny] == '.') {
68                     nswan.push(make_pair(nx,ny));
69                     scheck[nx][ny] = true;
70                 } else {
71                     nwater.push(make_pair(nx,ny));
72                     scheck[nx][ny] = true;
73                 }
74             }
75         }
76         if (scheck[ex][ey]) {
77             cout << i << '\n';
78             break;
79         }
80         water = nwater;
81         nswan = nwater;
82         nwater = queue<pair<int,int>>();
83         nswan = queue<pair<int,int>>();
84     }
85     return 0;
86 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

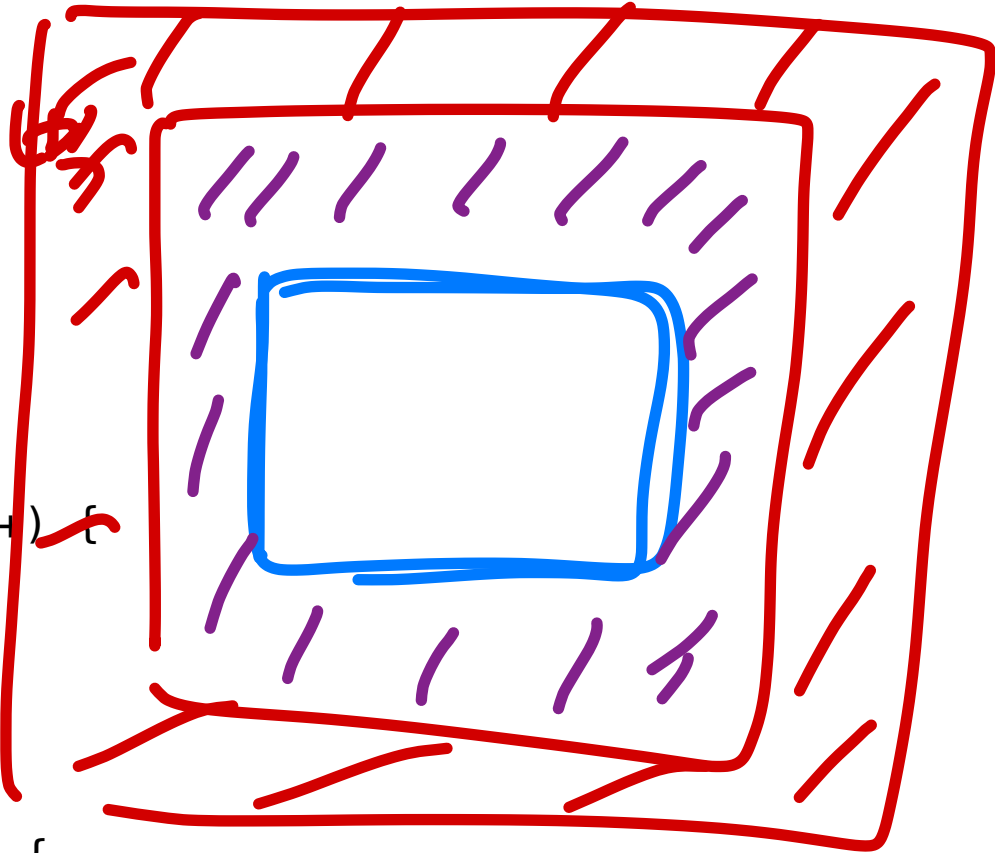
17852 KB

244 ms

2501 B

C++14

```
1 #include <stdio>
2 #include <cstring>
3 #include <queue>
4 using namespace std;
5 char a[111][111];
6 bool c[111][111];
7 bool key[111];
8 int dx[] = {0, 0, 1, -1};
9 int dy[] = {1, -1, 0, 0};
10 int main() {
11     int t;
12     scanf("%d",&t);
13     while (t--) {
14         int n, m;
15         memset(a,0,sizeof(a));
16         scanf("%d %d",&n,&m);
17         for (int i=2; i<n+2; i++) {
18             scanf("%s",a[i]+2);
19         }
20         n += 4;
21         m += 4;
22         for (int i=0; i<n; i++) {
23             a[i][0] = '*';
24             a[i][1] = '.';
25             a[i][m-2] = '.';
26             a[i][m-1] = '*';
27         }
28         for (int j=1; j<m-1; j++) {
29             a[0][j] = '*';
30             a[1][j] = '.';
31             a[n-2][j] = '.';
32             a[n-1][j] = '*';
33         }
34         memset(key,false,sizeof(key));
35         char temp[111];
36         scanf("%s",temp);
37         int len = strlen(temp);
38         if (temp[0] != '0') {
39             for (int i=0; i<len; i++) {
40                 key[temp[i]-'a'] = true;
41             }
42         }
43         int ans = 0;
44         memset(c,false,sizeof(c));
45         queue<pair<int,int>> q;
46         queue<pair<int,int>> door[26];
47         q.push(make_pair(1,1));
48         c[1][1] = true;
49         while (!q.empty()) {
50             int x = q.front().first;
51             int y = q.front().second;
52             q.pop();
53             for (int k=0; k<4; k++) {
54                 int nx = x+dx[k];
55                 int ny = y+dy[k];
56                 if (c[nx][ny]) {
57                     continue;
58                 }
59                 char w = a[nx][ny];
60                 if (w == '*') {
61                     continue;
62                 }
63                 c[nx][ny] = true;
64                 if (w == '.') {
65                     q.push(make_pair(nx,ny));
66                 } else if (w == '$') {
67                     ans += 1;
68                     q.push(make_pair(nx,ny));
69                 } else if (w >= 'A' && w <= 'Z') {
70                     if (key[w-'A']) {
71                         q.push(make_pair(nx,ny));
72                     } else {
73                         door[w-'A'].push(make_pair(nx,ny));
74                     }
75                 } else if (w >= 'a' && w <= 'z') {
76                     q.push(make_pair(nx,ny));
77                     if (!key[w-'a']) {
78                         key[w-'a'] = true;
79                         while (!door[w-'a'].empty()) {
80                             q.push(door[w-'a'].front());
81                             door[w-'a'].pop();
82                         }
83                     }
84                 }
85             }
86         }
87         printf("%d\n",ans);
88     }
89     return 0;
90 }
```



false

(x,y)

(nx,ny)

12

열쇠

DFS

BFS

최소 이동

연결성

결과

메모리

시간

코드 길이

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <tuple>
4 #include <queue>
5 using namespace std;
6 int d[1000][1000];
7 int a[1000][1000];
8 int s[10];
9 int dx[] = {0,0,1,-1};
10 int dy[] = {1,-1,0,0};
11 queue<pair<int,int>> q[10];
12 queue<pair<int,int>> next_q[10];
13 int main() {
14     int n, m, p;
15     cin >> n >> m >> p;
16     for (int i=1; i<=p; i++) {
17         cin >> s[i];
18     }
19     for (int i=0; i<n; i++) {
20         string line;
21         cin >> line;
22         for (int j=0; j<m; j++) {
23             if (line[j] == '.') {
24                 a[i][j] = 0;
25             } else if (line[j] == '#') {
26                 a[i][j] = -1;
27             } else {
28                 a[i][j] = line[j] - '0';
29             }
30         }
31     }
32     for (int i=0; i<n; i++) {
33         for (int j=0; j<m; j++) {
34             if (a[i][j] > 0) {
35                 q[a[i][j]].push(make_pair(i, j));
36             }
37         }
38     }
39     while (true) {
40         bool ok = false;
41         for (int who=1; who<=p; who++) {
42             memset(d, 0, sizeof(d));
43             while (!q[who].empty()) {
44                 ok = true;
45                 int x, y;
46                 tie(x, y) = q[who].front();
47                 q[who].pop();
48                 if (d[x][y] == s[who]) {
49                     next_q[who].push(make_pair(x, y));
50                 }
51                 if (a[x][y] > 0 && a[x][y] != who) {
52                     continue;
53                 }
54                 a[x][y] = who;
55                 for (int k=0; k<4; k++) {
56                     int nx = x+dx[k];
57                     int ny = y+dy[k];
58                     if (0 <= nx && nx < n && 0 <= ny && ny < m) {
59                         if (a[nx][ny] == 0) {
60                             d[nx][ny] = d[x][y] + 1;
61                             if (d[nx][ny] <= s[who]) {
62                                 a[nx][ny] = who;
63                                 q[who].push(make_pair(nx, ny));
64                             }
65                         }
66                     }
67                 }
68             }
69             q[who] = next_q[who];
70             next_q[who] = queue<pair<int,int>>();
71         }
72         if (!ok) {
73             break;
74         }
75     }
76     vector<int> ans(p+1);
77     for (int i=0; i<n; i++) {
78         for (int j=0; j<m; j++) {
79             if (a[i][j] > 0) {
80                 ans[a[i][j]] += 1;
81             }
82         }
83     }
84     for (int i=1; i<=p; i++) {
85         cout << ans[i] << ' ';
86     }
87     cout << '\n';
88     return 0;
89 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

17256 KB

168 ms

2393 B



C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <tuple>
4 #include <vector>
5 #include <string>
6 #include <cstring>
7 using namespace std;
8 int dx[] = {0,0,1,-1};
9 int dy[] = {1,-1,0,0};
10 int d[10][10][10][10];
11 int rx,ry;
12 pair<bool,bool> simulate(vector<string> &a, int k, int &x, int &y) {
13     if (a[x][y] == '.') return make_pair(false, false);
14     int n = a.size();
15     int m = a[0].size();
16     bool moved = false;
17     while (true) {
18         int nx = x+dx[k];
19         int ny = y+dy[k];
20         if (nx < 0 || nx >= n || ny < 0 || ny >= m) {
21             return make_pair(moved, false);
22         }
23         if (a[nx][ny] == '#') {
24             return make_pair(moved, false);
25         } else if (a[nx][ny] == 'R' || a[nx][ny] == 'B') {
26             return make_pair(moved, false);
27         } else if (a[nx][ny] == '.') {
28             swap(a[nx][ny], a[x][y]);
29             x = nx;
30             y = ny;
31             moved = true;
32         } else if (a[nx][ny] == 'O') {
33             a[x][y] = '.';
34             moved = true;
35             return make_pair(moved, true);
36         }
37     }
38     return make_pair(false, false);
39 }
40 pair<bool,bool> next(vector<string> a, int &rx, int &ry, int &bx, int &by, int dir) {
41     a[rx][ry] = 'R';
42     a[bx][by] = 'B';
43     bool hole1=false, hole2=false;
44     while (true) {
45         auto p1 = simulate(a, dir, rx, ry);
46         auto p2 = simulate(a, dir, bx, by);
47         if (p1.first == false && p2.first == false) {
48             break;
49         }
50         if (p1.second) hole1 = true;
51         if (p2.second) hole2 = true;
52     }
53     return make_pair(hole1, hole2);
54 }
55 int main() {
56     int n, m;
57     cin >> n >> m;
58     vector<string> a(n);
59     for (int i=0; i<n; i++) {
60         cin >> a[i];
61     }
62     int ans = -1;
63     queue<tuple<int,int,int,int>> q;
64     int rx,ry,bx,by;
65     for (int i=0; i<n; i++) {
66         for (int j=0; j<m; j++) {
67             if (a[i][j] == 'O') {
68                 hx = i; hy = j;
69             } else if (a[i][j] == 'R') {
70                 rx = i; ry = j;
71             } else if (a[i][j] == 'B') {
72                 bx = i; by = j;
73             }
74         }
75     }
76     memset(d, -1, sizeof(d));
77     q.emplace(rx,ry,bx,by);
78     d[rx][ry][bx][by] = 0;
79     bool found = false;
80     while (!q.empty()) {
81         tie(rx,ry,bx,by) = q.front();
82         q.pop();
83         for (int k=0; k<4; k++) {
84             bool hole1, hole2;
85             int nrx = rx, nry = ry, nbx = bx, nby = by;
86             tie(hole1,hole2) = next(a,nrx,nry,nbx,nby,k);
87             if (hole2) continue;
88             if (hole1) {
89                 found = true;
90                 ans = d[rx][ry][bx][by] + 1;
91                 break;
92             }
93             if (d[nrx][nry][nbx][nby] != -1) continue;
94             q.emplace(nrx,nry,nbx,nby);
95             d[nrx][nry][nbx][nby] = d[rx][ry][bx][by] + 1;
96         }
97         if (found) {
98             break;
99         }
100     }
101     cout << ans << '\n';
102     return 0;
103 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

2032 KB

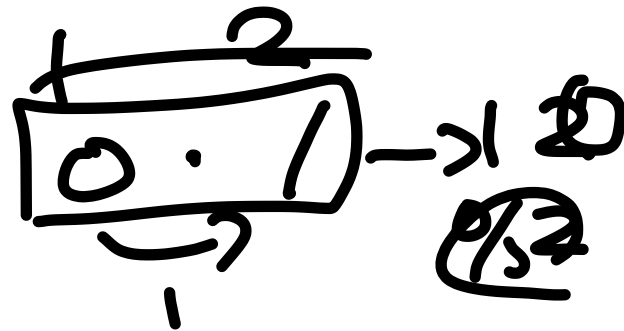
0 ms

2936 B



C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <queue>
4 #include <string>
5 #include <vector>
6 using namespace std;
7 int main() {
8     int n, k;
9     cin >> n >> k;
10    vector<string> a(2);
11    for (int i=0; i<2; i++) {
12        cin >> a[i];
13    }
14    vector<pair<int,int>> dirs = {{0,1},{0,-1},{1,k}};
15    vector<vector<int>> d(2, vector<int>(n,-1));
16    queue<pair<int,int>> q;
17    d[0][0] = 0;
18    q.push(make_pair(0,0));
19    bool ok = false;
20    while (!q.empty()) {
21        int x, y;
22        tie(x,y) = q.front(); q.pop();
23        for (auto &dir : dirs) {
24            int dx,dy;
25            tie(dx,dy) = dir;
26            int nx = (x+dx)%2;
27            int ny = y+dy;
28            if (ny >= n) {
29                ok = true;
30                break;
31            }
32            if (ny < 0) continue;
33            if (d[nx][ny] != -1) continue;
34            if (a[nx][ny] == '0') continue;
35            if (ny < d[x][y]+1) continue;
36            d[nx][ny] = d[x][y] + 1;
37            q.push(make_pair(nx,ny));
38        }
39        if (ok) break;
40    }
41    cout << (ok ? "1" : "0") << '\n';
42    return 0;
43 }
```



원문(기)

(0,0)

(x,y) → (nx,ny)

ny이  
값을 갖는  
최소

↓ 값 [ny] ny초

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](mailto:codeplus@startlink.io) 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.