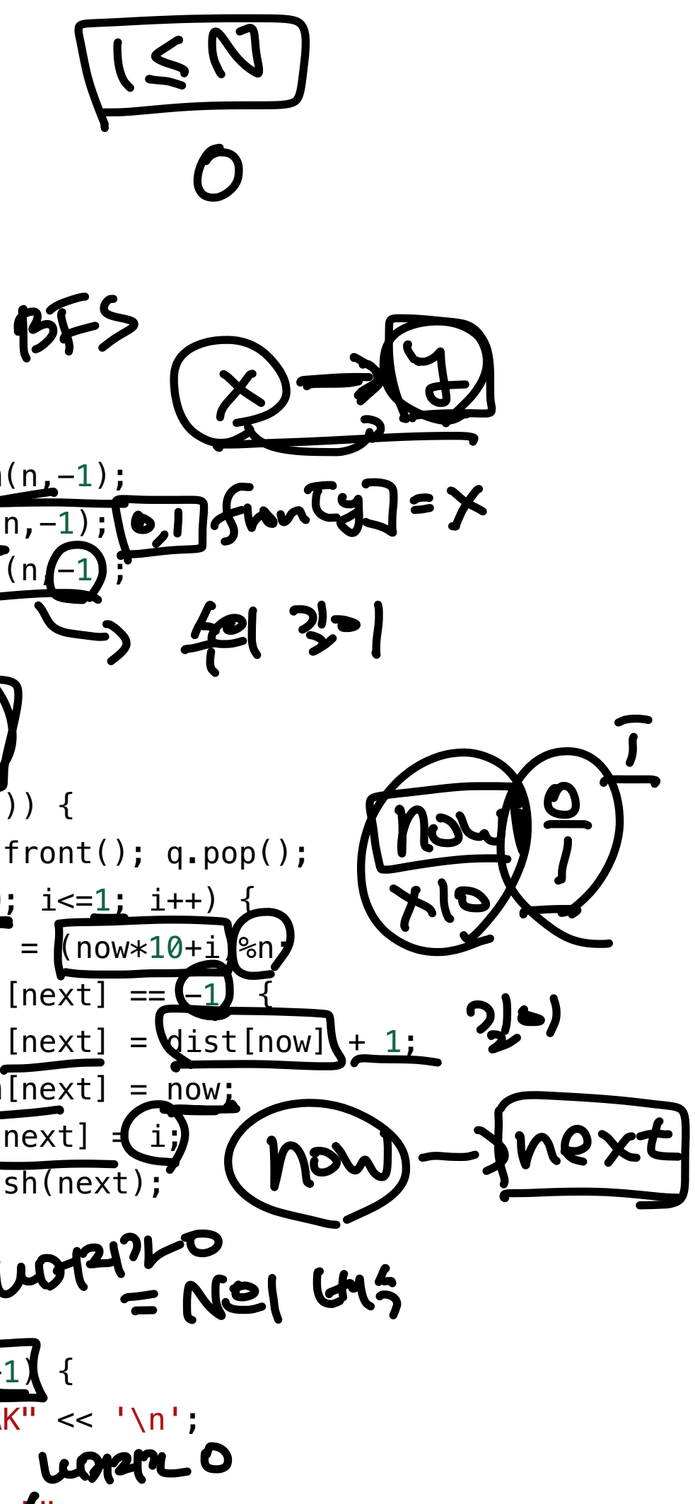
BFS 2

쇼스코드

최백준 choi@startlink.io

C++14

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <queue>
using namespace std;
int main() {
    int t;
    cin >> t;
    while (t--) {
        int n;
        cin >> n;
        vector<int> from(n,-1);
        vector<int> how(n,-1);
        vector<int> dist(n,-1);
        queue<int> q;
        q.push(1%n);
        dist[1%n] = 0;
        how[1%n] = 1;
        while (!q.empty()) {
            int now = q.front(); q.pop();
            for (int i=0; i<=1; i++) {
                int next = (now*10+i)%n;
                if (dist[next] == -1) {
                    dist[next] = dist[now] + 1;
                    from[next] = now;
                    how[next] = i;
                    q.push(next);
                }
            }
        }
        if (dist[0] == -1) {
            cout << "BRAK" << '\n';
        } else {
            string ans = "";
            for (int i=0; i!=-1; i=from[i]) {
                ans += to_string(how[i]);
            }
            reverse(ans.begin(),ans.end());
            cout << ans << '\n';
        }
    }
    return 0;
}
```



결과	메모리	시간	코드 길이
맞았습니다!!	2180 KB	0 ms	1096 B

C++14

```cpp
#include <iostream>
#include <cstring>
#include <queue>
using namespace std;
int dist[500001];
int main() {
    int n, k;
    cin >> n >> k;
    if (n == k) {
        cout << 0 << '\n';
        return 0;
    }
    memset(dist,-1,sizeof(dist));
    queue<int> q;
    q.push(n);
    dist[n] = 0;
    for (int t=1;; t++) {
        k += t;
        if (k > 500000) break;
        queue<int> nq;
        while (!q.empty()) {
            int x = q.front();
            q.pop();
            for (int y : {x+1, x-1, 2*x}) {
                if (0 <= y && y <= 500000) {
                    if (dist[y] != t) {
                        nq.push(y);
                        dist[y] = t;
                    }
                }
            }
        }
        if (dist[k] == t) {
            cout << t << '\n';
            return 0;
        }
        q = nq;
    }
    cout << -1 << '\n';
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	7756 KB	4960 ms	876 B

C++14

```cpp
#include <iostream>
#include <tuple>
#include <cstring>
#include <queue>
using namespace std;
int dist[500001][2];
int main() {
    int n, k;
    cin >> n >> k;
    memset(dist,-1,sizeof(dist));
    queue<pair<int,int>> q;
    q.push(make_pair(n, 0));
    dist[n][0] = 0;
    while (!q.empty()) {
        int x, t;
        tie(x, t) = q.front(); q.pop();
        for (int y : {x+1, x-1, 2*x}) {
            if (0 <= y && y <= 500000) {
                if (dist[y][1-t] == -1) {
                    dist[y][1-t] = dist[x][t] + 1;
                    q.push(make_pair(y, 1-t));
                }
            }
        }
    }
    int ans = -1;
    for (int t=0;; t++) {
        k += t;
        if (k > 500000) break;
        if (dist[k][t%2] <= t) {
            ans = t;
            break;
        }
    }
    cout << ans << '\n';
    return 0;
}
```

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

C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
using namespace std;
int a[1001][1001];
int d[1001][1001];
int s[1001][1001];
int dx[] = {1,-1,0,0};
int dy[] = {0,0,1,-1};
int sum(int x1, int y1, int x2, int y2) {
    return s[x2][y2] - s[x1-1][y2] - s[x2][y1-1] + s[x1-1][y1-1];
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n, m;
    cin >> n >> m;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=m; j++) {
            cin >> a[i][j];
            d[i][j] = -1;
        }
    }
    int h, w;
    cin >> h >> w;
    int x1, y1, x2, y2;
    cin >> x1 >> y1 >> x2 >> y2;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=m; j++) {
            s[i][j] = s[i-1][j] + s[i][j-1] - s[i-1][j-1] + a[i][j];
        }
    }
    queue<pair<int,int>> q;
    q.push(make_pair(x1,y1));
    d[x1][y1] = 0;
    while (!q.empty()) {
        int x, y;
        tie(x, y) = q.front(); q.pop();
        for (int k=0; k<4; k++) {
            int nx = x+dx[k];
            int ny = y+dy[k];
            if (1 <= nx && nx+h-1 <= n && 1 <= ny && ny+w-1 <= m) {
                if (sum(nx,ny,nx+h-1,ny+w-1) == 0) {
                    if (d[nx][ny] == -1) {
                        q.push(make_pair(nx,ny));
                        d[nx][ny] = d[x][y] + 1;
                    }
                }
            }
        }
    }
    cout << d[x2][y2] << '\n';
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	13860 KB	104 ms	1405 B

C++14

```cpp
#include <iostream>
#include <tuple>
#include <cstring>
#include <queue>
#include <vector>
#include <string>
using namespace std;
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int d[51][51][4][4];
int main() {
    int n, m;
    cin >> n >> m;
    vector<string> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    int ans = -1;
    int x1,y1,x2,y2;
    x1 = y1 = x2 = y2 = -1;
    memset(d,-1,sizeof(d));
    queue<tuple<int,int,int,int>> q;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 'S') {
                for (int k=0; k<4; k++) {
                    q.push(make_tuple(i,j,k,0));
                    d[i][j][k][0] = 0;
                }
            } else if (a[i][j] == 'C') {
                if (x1 == -1) {
                    x1 = i;
                    y1 = j;
                } else {
                    x2 = i;
                    x2 = j;
                }
            }
        }
    }
    while (!q.empty()) {
        int x,y,dir,s;
        tie(x,y,dir,s) = q.front();
        q.pop();
        if (s == 3) {
            ans = d[x][y][dir][s];
            break;
        }
        for (int k=0; k<4; k++) {
            if (dir == k) continue;
            int nx = x+dx[k];
            int ny = y+dy[k];
            if (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (a[nx][ny] != '#') {
                    int ns = s;
                    if (a[nx][ny] == 'C') {
                        if (nx == x1 && ny == y1) {
                            ns |= 1;
                        } else {
                            ns |= 2;
                        }
                    }
                    if (d[nx][ny][k][ns] == -1) {
                        d[nx][ny][k][ns] = d[x][y][dir][s] + 1;
                        q.push(make_tuple(nx,ny,k,ns));
                    }
                }
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	2156 KB	0 ms	1946 B

C++14

```cpp
#include <iostream>
#include <cstring>
#include <tuple>
#include <queue>
#include <vector>
using namespace std;
int n;
int a[10][10];
int d[10][10][100][3]; // (x, y, num, piece)
queue<tuple<int,int,int,int>> q;
int dx1[] = {-2,-1,1,2,2,1,-1,-2};
int dy1[] = {1,2,2,1,-1,-2,-2,-1};
int dx2[] = {0,0,1,-1};
int dy2[] = {1,-1,0,0};
int dx3[] = {1,1,-1,-1};
int dy3[] = {1,-1,1,-1};
int main() {
    memset(d,-1,sizeof(d));
    cin >> n;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            a[i][j] -= 1;
            if (a[i][j] == 0) {
                for (int k=0; k<3; k++) {
                    d[i][j][0][k] = 0;
                    q.push(make_tuple(i,j,0,k));
                }
            }
        }
    }
    int ans = -1;
    while (!q.empty()) {
        int x, y, num, piece;
        tie(x, y, num, piece) = q.front(); q.pop();
        if (num == n*n-1) {
            if (ans == -1 || ans > d[x][y][num][piece]) {
                ans = d[x][y][num][piece];
            }
        }
        for (int i=0; i<3; i++) {
            if (piece == i) continue;
            if (d[x][y][num][i] == -1) {
                d[x][y][num][i] = d[x][y][num][piece] + 1;
                q.push(make_tuple(x,y,num,i));
            }
        }
        if (piece == 0) { // knight
            for (int k=0; k<8; k++) {
                int nx = x+dx1[k];
                int ny = y+dy1[k];
                if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                    int next_num = num;
                    if (a[nx][ny] == num+1) {
                        next_num = num+1;
                    }
                    if (d[nx][ny][next_num][piece] == -1) {
                        d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
                        q.push(make_tuple(nx, ny, next_num, piece));
                    }
                }
            }
        } else if (piece == 1) { // rook
            for (int k=0; k<4; k++) {
                for (int l=1; l<n; l++) {
                    int nx = x+dx2[k]*l;
                    int ny = y+dy2[k]*l;
                    if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                        int next_num = num;
                        if (a[nx][ny] == num+1) {
                            next_num = num+1;
                        }
                        if (d[nx][ny][next_num][piece] == -1) {
                            d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
                            q.push(make_tuple(nx, ny, next_num, piece));
                        }
                    } else {
                        break;
                    }
                }
            }
        } else { // bishop
            for (int k=0; k<4; k++) {
                for (int l=1; l<n; l++) {
                    int nx = x+dx3[k]*l;
                    int ny = y+dy3[k]*l;
                    if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                        int next_num = num;
                        if (a[nx][ny] == num+1) {
                            next_num = num+1;
                        }
                        if (d[nx][ny][next_num][piece] == -1) {
                            d[nx][ny][next_num][piece] = d[x][y][num][piece] + 1;
                            q.push(make_tuple(nx, ny, next_num, piece));
                        }
                    } else {
                        break;
                    }
                }
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	2104 KB	4 ms	3525 B

C++14

```cpp
#include <iostream>
#include <cstring>
#include <tuple>
#include <queue>
#include <vector>
using namespace std;
int n;
int a[10][10];
pair<int,int> d[10][10][100][3]; // (x, y, num, piece)
queue<tuple<int,int,int,int>> q;
int dx1[] = {-2,-1,1,2,2,1,-1,-2};
int dy1[] = {1,2,2,1,-1,-2,-2,-1};
int dx2[] = {0,0,1,-1};
int dy2[] = {1,-1,0,0};
int dx3[] = {1,1,-1,-1};
int dy3[] = {1,-1,1,-1};
int main() {
    memset(d,-1,sizeof(d));
    cin >> n;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            for (int k=0; k<n*n; k++) {
                for (int l=0; l<3; l++) {
                    d[i][j][k][l] = make_pair(-1,-1);
                }
            }
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            a[i][j] -= 1;
            if (a[i][j] == 0) {
                for (int k=0; k<3; k++) {
                    d[i][j][0][k] = make_pair(0,0);
                    q.push(make_tuple(i,j,0,k));
                }
            }
        }
    }
    pair<int,int> ans = make_pair(-1,-1);
    while (!q.empty()) {
        int x, y, num, piece;
        tie(x, y, num, piece) = q.front(); q.pop();
        auto &p = d[x][y][num][piece];
        if (num == n*n-1) {
            if (ans.first == -1 || ans > p) {
                ans = p;
            }
        }
        for (int i=0; i<3; i++) {
            if (piece == i) continue;
            auto np = make_pair(p.first+1, p.second+1);
            if (d[x][y][num][i].first == -1 || d[x][y][num][i] > np) {
                d[x][y][num][i] = np;
                q.push(make_tuple(x,y,num,i));
            }
        }
        if (piece == 0) { // knight
            for (int k=0; k<8; k++) {
                int nx = x+dx1[k];
                int ny = y+dy1[k];
                if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                    int next_num = num;
                    if (a[nx][ny] == num+1) {
                        next_num = num+1;
                    }
                    auto np = make_pair(p.first+1, p.second);
                    if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
                        d[nx][ny][next_num][piece] = np;
                        q.push(make_tuple(nx, ny, next_num, piece));
                    }
                }
            }
        } else if (piece == 1) { // rook
            for (int k=0; k<4; k++) {
                for (int l=1;; l++) {
                    int nx = x+dx2[k]*l;
                    int ny = y+dy2[k]*l;
                    if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                        int next_num = num;
                        if (a[nx][ny] == num+1) {
                            next_num = num+1;
                        }
                        auto np = make_pair(p.first+1, p.second);
                        if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
                            d[nx][ny][next_num][piece] = np;
                            q.push(make_tuple(nx, ny, next_num, piece));
                        }
                    } else {
                        break;
                    }
                }
            }
        } else { // bishop
            for (int k=0; k<4; k++) {
                for (int l=1;; l++) {
                    int nx = x+dx3[k]*l;
                    int ny = y+dy3[k]*l;
                    if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                        int next_num = num;
                        if (a[nx][ny] == num+1) {
                            next_num = num+1;
                        }
                        auto np = make_pair(p.first+1, p.second);
                        if (d[nx][ny][next_num][piece].first == -1 || d[nx][ny][next_num][piece] > np) {
                            d[nx][ny][next_num][piece] = np;
                            q.push(make_tuple(nx, ny, next_num, piece));
                        }
                    } else {
                        break;
                    }
                }
            }
        }
    }
    cout << ans.first << ' ' << ans.second << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	2224 KB	4 ms	4175 B

C++14

```cpp
#include <iostream>
#include <vector>
#include <queue>
#include <string>
using namespace std;
string a[1500];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
bool wcheck[1500][1500];
bool scheck[1500][1500];
int main() {
    int n, m;
    cin >> n >> m;
    int sx,sy,ex,ey;
    sx=sy=ex=ey=-1;
    queue<pair<int,int>> swan, nswan, water, nwater;
    for (int i=0; i<n; i++) {
        cin >> a[i];
        for (int j=0; j<m; j++) {
            if (a[i][j] == 'L') {
                if (sx == -1) {
                    sx = i;
                    sy = j;
                } else {
                    ex = i;
                    ey = j;
                }
                a[i][j] = '.';
            }
            if (a[i][j] == '.') {
                water.push(make_pair(i,j));
                wcheck[i][j] = true;
            }
        }
    }
    swan.push(make_pair(sx,sy));
    scheck[sx][sy] = true;
    for (int i=0; ; i++) {
        while (!water.empty()) {
            int x = water.front().first;
            int y = water.front().second;
            water.pop();
            a[x][y] = '.';
            for (int k=0; k<4; k++) {
                int nx = x+dx[k];
                int ny = y+dy[k];
                if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
                if (wcheck[nx][ny]) continue;
                if (a[nx][ny] == '.') {
                    water.push(make_pair(nx,ny));
                    wcheck[nx][ny] = true;
                } else {
                    nwater.push(make_pair(nx,ny));
                    wcheck[nx][ny] = true;
                }
            }
        }
        while (!swan.empty()) {
            int x = swan.front().first;
            int y = swan.front().second;
            swan.pop();
            for (int k=0; k<4; k++) {
                int nx = x+dx[k];
                int ny = y+dy[k];
                if (nx < 0 || nx >= n || ny < 0 || ny >= m) continue;
                if (scheck[nx][ny]) continue;
                if (a[nx][ny] == '.') {
                    swan.push(make_pair(nx,ny));
                    scheck[nx][ny] = true;
                } else {
                    nswan.push(make_pair(nx,ny));
                    scheck[nx][ny] = true;
                }
            }
        }
        if (scheck[ex][ey]) {
            cout << i << '\n';
            break;
        }
        water = nwater;
        swan = nswan;
        nwater = queue<pair<int,int>>();
        nswan = queue<pair<int,int>>();
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	17852 KB	244 ms	2501 B

C++14

```cpp
#include <cstdio>
#include <cstring>
#include <queue>
using namespace std;
char a[111][111];
bool c[111][111];
bool key[111];
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int main() {
    int t;
    scanf("%d",&t);
    while (t--) {
        int n, m;
        memset(a,0,sizeof(a));
        scanf("%d %d",&n,&m);
        for (int i=2; i<n+2; i++) {
            scanf("%s",a[i]+2);
        }
        n += 4;
        m += 4;
        for (int i=0; i<n; i++) {
            a[i][0] = '.';
            a[i][1] = '.';
            a[i][m-2] = '.';
            a[i][m-1] = '.';
        }
        for (int j=1; j<m-1; j++) {
            a[0][j] = '.';
            a[1][j] = '.';
            a[n-2][j] = '.';
            a[n-1][j] = '.';
        }
        memset(key,false,sizeof(key));
        char temp[111];
        scanf("%s",temp);
        int len = strlen(temp);
        if (temp[0] != '0') {
            for (int i=0; i<len; i++) {
                key[temp[i]-'a'] = true;
            }
        }
        int ans = 0;
        memset(c,false,sizeof(c));
        queue<pair<int,int>> q;
        queue<pair<int,int>> door[26];
        q.push(make_pair(1,1));
        c[1][1] = true;
        while (!q.empty()) {
            int x = q.front().first;
            int y = q.front().second;
            q.pop();
            for (int k=0; k<4; k++) {
                int nx = x+dx[k];
                int ny = y+dy[k];
                if (c[nx][ny]) {
                    continue;
                }
                char w = a[nx][ny];
                if (w == '*') {
                    continue;
                }
                c[nx][ny] = true;
                if (w == '.') {
                    q.push(make_pair(nx,ny));
                } else if (w == '$') {
                    ans += 1;
                    q.push(make_pair(nx,ny));
                } else if (w >= 'A' && w <= 'Z') {
                    if (key[w-'A']) {
                        q.push(make_pair(nx,ny));
                    } else {
                        door[w-'A'].push(make_pair(nx,ny));
                    }
                } else if (w >= 'a' && w <= 'z') {
                    q.push(make_pair(nx,ny));
                    if (!key[w-'a']) {
                        key[w-'a'] = true;
                        while (!door[w-'a'].empty()) {
                            q.push(door[w-'a'].front());
                            door[w-'a'].pop();
                        }
                    }
                }
            }
        }
        printf("%d\n",ans);
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	1252 KB	4 ms	2564 B

C++14

```cpp
#include <iostream>
#include <cstring>
#include <tuple>
#include <queue>
using namespace std;
int d[1000][1000];
int a[1000][1000];
int s[10];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
queue<pair<int,int>> q[10];
queue<pair<int,int>> next_q[10];
int main() {
    int n, m, p;
    cin >> n >> m >> p;
    for (int i=1; i<=p; i++) {
        cin >> s[i];
    }
    for (int i=0; i<n; i++) {
        string line;
        cin >> line;
        for (int j=0; j<m; j++) {
            if (line[j] == '.') {
                a[i][j] = 0;
            } else if (line[j] == '#') {
                a[i][j] = -1;
            } else {
                a[i][j] = line[j]-'0';
            }
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] > 0) {
                q[a[i][j]].push(make_pair(i, j));
            }
        }
    }
    while (true) {
        bool ok = false;
        for (int who=1; who<=p; who++) {
            memset(d,0,sizeof(d));
            while (!q[who].empty()) {
                ok = true;
                int x, y;
                tie(x, y) = q[who].front();
                q[who].pop();
                if (d[x][y] == s[who]) {
                    next_q[who].push(make_pair(x,y));
                }
                if (a[x][y] > 0 && a[x][y] != who) {
                    continue;
                }
                a[x][y] = who;
                for (int k=0; k<4; k++) {
                    int nx = x+dx[k];
                    int ny = y+dy[k];
                    if (0 <= nx && nx < n && 0 <= ny && ny < m) {
                        if (a[nx][ny] == 0) {
                            d[nx][ny] = d[x][y] + 1;
                            if (d[nx][ny] <= s[who]) {
                                a[nx][ny] = who;
                                q[who].push(make_pair(nx,ny));
                            }
                        }
                    }
                }
            }
            q[who] = next_q[who];
            next_q[who] = queue<pair<int,int>>();
        }
        if (!ok) {
            break;
        }
    }
    vector<int> ans(p+1);
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] > 0) {
                ans[a[i][j]] += 1;
            }
        }
    }
    for (int i=1; i<=p; i++) {
        cout << ans[i] << ' ';
    }
    cout << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	17256 KB	168 ms	2393 B

C++14

```cpp
#include <iostream>
#include <queue>
#include <tuple>
#include <vector>
#include <string>
#include <cstring>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int d[10][10][10][10];
int hx,hy;
pair<bool,bool> simulate(vector<string> &a, int k, int &x, int &y) {
    if (a[x][y] == '.') return make_pair(false, false);
    int n = a.size();
    int m = a[0].size();
    bool moved = false;
    while (true) {
        int nx = x+dx[k];
        int ny = y+dy[k];
        if (nx < 0 || nx >= n || ny < 0 || ny >= m) {
            return make_pair(moved, false);
        }
        if (a[nx][ny] == '#') {
            return make_pair(moved, false);
        } else if (a[nx][ny] == 'R' || a[nx][ny] == 'B') {
            return make_pair(moved, false);
        } else if (a[nx][ny] == '.') {
            swap(a[nx][ny], a[x][y]);
            x = nx;
            y = ny;
            moved = true;
        } else if (a[nx][ny] == 'O') {
            a[x][y] = '.';
            moved = true;
            return make_pair(moved, true);
        }
    }
    return make_pair(false, false);
}
pair<bool,bool> next(vector<string> a, int &rx, int &ry, int &bx, int &by, int dir) {
    a[rx][ry] = 'R';
    a[bx][by] = 'B';
    bool hole1=false, hole2=false;
    while (true) {
        auto p1 = simulate(a, dir, rx, ry);
        auto p2 = simulate(a, dir, bx, by);
        if (p1.first == false && p2.first == false) {
            break;
        }
        if (p1.second) hole1 = true;
        if (p2.second) hole2 = true;
    }
    return make_pair(hole1, hole2);
}
int main() {
    int n, m;
    cin >> n >> m;
    vector<string> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    int ans = -1;
    queue<tuple<int,int,int,int>> q;
    int rx,ry,bx,by;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 'O') {
                hx = i; hy = j;
            } else if (a[i][j] == 'R') {
                rx = i; ry = j;
                a[i][j] = '.';
            } else if (a[i][j] == 'B') {
                bx = i; by = j;
                a[i][j] = '.';
            }
        }
    }
    memset(d,-1,sizeof(d));
    q.emplace(rx,ry,bx,by);
    d[rx][ry][bx][by] = 0;
    bool found = false;
    while (!q.empty()) {
        tie(rx,ry,bx,by) = q.front();
        q.pop();
        for (int k=0; k<4; k++) {
            bool hole1, hole2;
            int nrx = rx, nry = ry, nbx = bx, nby = by;
            tie(hole1,hole2) = next(a,nrx,nry,nbx,nby,k);
            if (hole2) continue;
            if (hole1) {
                found = true;
                ans = d[rx][ry][bx][by] + 1;
                break;
            }
            if (d[nrx][nry][nbx][nby] != -1) continue;
            q.emplace(nrx,nry,nbx,nby);
            d[nrx][nry][nbx][nby] = d[rx][ry][bx][by] + 1;
        }
        if (found) {
            break;
        }
    }
    cout << ans << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	2032 KB	0 ms	2936 B

C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
#include <string>
#include <vector>
using namespace std;
int main() {
    int n, k;
    cin >> n >> k;
    vector<string> a(2);
    for (int i=0; i<2; i++) {
        cin >> a[i];
    }
    vector<pair<int,int>> dirs = {{0,1},{0,-1},{1,k}};
    vector<vector<int>> d(2, vector<int>(n, -1));
    queue<pair<int,int>> q;
    d[0][0] = 0;
    q.push(make_pair(0,0));
    bool ok = false;
    while (!q.empty()) {
        int x, y;
        tie(x,y) = q.front(); q.pop();
        for (auto &dir : dirs) {
            int dx,dy;
            tie(dx,dy) = dir;
            int nx = (x+dx)%2;
            int ny = y+dy;
            if (ny >= n) {
                ok = true;
                break;
            }
            if (ny < 0) continue;
            if (d[nx][ny] != -1) continue;
            if (a[nx][ny] == '0') continue;
            if (ny < d[x][y]+1) continue;
            d[nx][ny] = d[x][y] + 1;
            q.push(make_pair(nx,ny));
        }
        if (ok) break;
    }
    cout << (ok ? "1" : "0") << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	3412 KB	12 ms	1085 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.