# BFS (인접) 소스코드

최백준 choi@startlink.io

## C++14

```cpp
#include <iostream>
#include <algorithm>
#include <queue>
#define next _next
using namespace std;
int dist[101];
int next[101];
int main() {
    int n, m;
    cin >> n >> m;
    for (int i=1; i<=100; i++) {
        next[i] = i;
        dist[i] = -1;
    }
    while (n--) {
        int x, y;
        cin >> x >> y;
        next[x] = y;
    }
    while (m--) {
        int x, y;
        cin >> x >> y;
        next[x] = y;
    }
    dist[1] = 0;
    queue<int> q;
    q.push(1);
    while (!q.empty()) {
        int x = q.front(); q.pop();
        for (int i=1; i<=6; i++) {
            int y = x+i;
            if (y > 100) continue;
            y = next[y];
            if (dist[y] == -1) {
                dist[y] = dist[x] + 1;
                q.push(y);
            }
        }
    }
    cout << dist[100] << '\n';
    return 0;
}

```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 1988 KB | 0 ms | 835 B |

## C++14

```cpp
#include <iostream>
#include <tuple>
#include <cstring>
#include <queue>
using namespace std;
int dx[] = {-2,-2,0,0,2,2};
int dy[] = {-1,1,-2,2,-1,1};
int dist[200][200];
int main() {
    int n;
    cin >> n;
    int sx,sy,ex,ey;
    cin >> sx >> sy >> ex >> ey;
    memset(dist,-1,sizeof(dist));
    dist[sx][sy] = 0;
    queue<pair<int,int>> q;
    q.push(make_pair(sx,sy));
    while (!q.empty()) {
        int x, y;
        tie(x, y) = q.front(); q.pop();
        for (int k=0; k<6; k++) {
            int nx = x+dx[k];
            int ny = y+dy[k];
            if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                if (dist[nx][ny] == -1) {
                    q.push(make_pair(nx,ny));
                    dist[nx][ny] = dist[x][y] + 1;
                }
            }
        }
    }
    cout << dist[ex][ey] << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 2144 KB | 0 ms | 850 B |

## C++14

```cpp
#include <iostream>
#include <queue>
using namespace std;
int n, m;
int a[10][10];
int b[10][10];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int bfs() {
    queue<pair<int,int>> q;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            b[i][j] = a[i][j];
            if (b[i][j] == 2) {
                q.push(make_pair(i,j));
            }
        }
    }
    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 (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (b[nx][ny] == 0) {
                    b[nx][ny] = 2;
                    q.push(make_pair(nx,ny));
                }
            }
        }
    }
    int cnt = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (b[i][j] == 0) {
                cnt += 1;
            }
        }
    }
    return cnt;
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    int ans = 0;
    for (int x1=0; x1<n; x1++) {
        for (int y1=0; y1<m; y1++) {
            if (a[x1][y1] != 0) continue;
            for (int x2=0; x2<n; x2++) {
                for (int y2=0; y2<m; y2++) {
                    if (a[x2][y2] != 0) continue;
                    for (int x3=0; x3<n; x3++) {
                        for (int y3=0; y3<m; y3++) {
                            if (a[x3][y3] != 0) continue;
                            if (x1 == x2 && y1 == y2) continue;
                            if (x1 == x3 && y1 == y3) continue;
                            if (x2 == x3 && y2 == y3) continue;
                            a[x1][y1] = 1;
                            a[x2][y2] = 1;
                            a[x3][y3] = 1;
                            int cur = bfs();
                            if (ans < cur) ans = cur;
                            a[x1][y1] = 0;
                            a[x2][y2] = 0;
                            a[x3][y3] = 0;
                        }
                    }
                }
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 1988 KB | 256 ms | 2246 B |

## C++14

```cpp
#include <iostream>
#include <queue>
using namespace std;
int n, m;
int a[10][10];
int b[10][10];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
void dfs(int x, int y) {
    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 (b[nx][ny] == 0) {
                b[nx][ny] = 2;
                dfs(nx,ny);
            }
        }
    }
}
int dfs() {
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            b[i][j] = a[i][j];
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (b[i][j] == 2) {
                dfs(i, j);
            }
        }
    }
    int cnt = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (b[i][j] == 0) {
                cnt += 1;
            }
        }
    }
    return cnt;
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    int ans = 0;
    for (int x1=0; x1<n; x1++) {
        for (int y1=0; y1<m; y1++) {
            if (a[x1][y1] != 0) continue;
            for (int x2=0; x2<n; x2++) {
                for (int y2=0; y2<m; y2++) {
                    if (a[x2][y2] != 0) continue;
                    for (int x3=0; x3<n; x3++) {
                        for (int y3=0; y3<m; y3++) {
                            if (a[x3][y3] != 0) continue;
                            if (x1 == x2 && y1 == y2) continue;
                            if (x1 == x3 && y1 == y3) continue;
                            if (x2 == x3 && y2 == y3) continue;
                            a[x1][y1] = 1;
                            a[x2][y2] = 1;
                            a[x3][y3] = 1;
                            int cur = dfs();
                            if (ans < cur) ans = cur;
                            a[x1][y1] = 0;
                            a[x2][y2] = 0;
                            a[x3][y3] = 0;
                        }
                    }
                }
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 1988 KB | 256 ms | 2143 B |

## C++14

```cpp
#include <iostream>
#include <queue>
using namespace std;
bool check[1501][1501];
int sum;
void go(int x, int y) {
    if (check[x][y]) return;
    check[x][y] = true;
    int a[3] = {x, y, sum-x-y};
    for (int i=0; i<3; i++) {
        for (int j=0; j<3; j++) {
            if (a[i] < a[j]) {
                int b[3] = {x, y, sum-x-y};
                b[i] += a[i];
                b[j] -= a[i];
                go(b[0], b[1]);
            }
        }
    }
}
int main() {
    int x, y, z;
    cin >> x >> y >> z;
    sum = x + y + z;
    if (sum % 3 != 0) {
        cout << 0 << '\n';
        return 0;
    }
    go(x, y);
    if (check[sum/3][sum/3]) {
        cout << 1 << '\n';
    } else {
        cout << 0 << '\n';
    }
    return 0;
}

```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 32688 KB | 52 ms | 747 B |

## C++14

```cpp
#include <iostream>
#include <queue>
#include <cstdio>
#include <tuple>
using namespace std;
int a[1000][1000];
int d[1000][1000][2];
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int main() {
    int n, m;
    scanf("%d %d",&n,&m);
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            scanf("%1d",&a[i][j]);
        }
    }
    queue<tuple<int,int,int>> q;
    d[0][0][0] = 1;
    q.push(make_tuple(0,0,0));
    while (!q.empty()) {
        int x, y, z;
        tie(x,y,z) = q.front(); q.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 (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));
            }
        }
    }
    if (d[n-1][m-1][0] != 0 && d[n-1][m-1][1] != 0) {
        cout << min(d[n-1][m-1][0], d[n-1][m-1][1]);
    } else if (d[n-1][m-1][0] != 0) {
        cout << d[n-1][m-1][0];
    } else if (d[n-1][m-1][1] != 0) {
        cout << d[n-1][m-1][1];
    } else {
        cout << -1;
    }
    cout << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 13708 KB | 108 ms | 1358 B |

## C++14

```cpp
#include <iostream>
#include <vector>
#include <string>
#include <queue>
#include <set>
#include <tuple>
using namespace std;
int n, m;
int a[1000][1000];
bool check[1000][1000];
int group[1000][1000];
vector<int> group_size;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
void bfs(int sx, int sy) {
    int g = group_size.size();
    queue<pair<int,int>> q;
    q.push(make_pair(sx,sy));
    check[sx][sy] = true;
    group[sx][sy] = g;
    int cnt = 1;
    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 (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (a[nx][ny] == 0 && check[nx][ny] == false) {
                    q.push(make_pair(nx,ny));
                    check[nx][ny] = true;
                    group[nx][ny] = g;
                    cnt += 1;
                }
            }
        }
    }
    group_size.push_back(cnt);
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        string s;
        cin >> s;
        for (int j=0; j<m; j++) {
            a[i][j] = s[j] - '0';
            check[i][j] = false;
            group[i][j] = -1;
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0 && check[i][j] == false) {
                bfs(i, j);
            }
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0) {
                cout << 0;
            } else {
                set<int> near;
                for (int k=0; k<4; k++) {
                    int x = i+dx[k];
                    int y = j+dy[k];
                    if (0 <= x && x < n && 0 <= y && y < m) {
                        if (a[x][y] == 0) {
                            near.insert(group[x][y]);
                        }
                    }
                }
                int ans = 1;
                for (int g : near) {
                    ans += group_size[g];
                }
                cout << ans%10;
            }
        }
        cout << '\n';
    }
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 11176 KB | 160 ms | 2156 B |

## C++14

```cpp
#include <iostream>
#include <queue>
#include <cstdio>
#include <tuple>
#include <cstring>
using namespace std;
int a[1000][1000];
int d[1000][1000][11];
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int main() {
    int n, m, l;
    scanf("%d %d %d",&n,&m, &l);
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            scanf("%1d",&a[i][j]);
        }
    }
    queue<tuple<int,int,int>> q;
    d[0][0][0] = 1;
    q.push(make_tuple(0,0,0));
    while (!q.empty()) {
        int x, y, z;
        tie(x,y,z) = q.front(); q.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 (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+1 <= l && 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));
            }
        }
    }
    int ans = -1;
    for (int i=0; i<=l; i++) {
        if (d[n-1][m-1][i] == 0) continue;
        if (ans == -1) {
            ans = d[n-1][m-1][i];
        } else if (ans > d[n-1][m-1][i]) {
            ans = d[n-1][m-1][i];
        }
    }
    cout << ans << '\n';
    return 0;
}

```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 48996 KB | 332 ms | 1356 B |

## C++14

```cpp
#include <iostream>
#include <queue>
#include <cstdio>
#include <tuple>
#include <cstring>
using namespace std;
int a[1000][1000];
int d[1000][1000][11][2];
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int main() {
    int n, m, l;
    scanf("%d %d %d",&n,&m, &l);
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            scanf("%1d",&a[i][j]);
        }
    }
    queue<tuple<int,int,int,int>> q;
    d[0][0][0][0] = 1;
    q.push(make_tuple(0,0,0,0));
    while (!q.empty()) {
        int x, y, z, night;
        tie(x,y,z, night) = q.front(); q.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 (a[nx][ny] == 0 && d[nx][ny][z][1-night] == 0) {
                d[nx][ny][z][1-night] = d[x][y][z][night] + 1;
                q.push(make_tuple(nx,ny,z,1-night));
            }
            if (night == 0 && z+1 <= l && a[nx][ny] == 1 && d[nx][ny][z+1][1-night] == 0) {
                d[nx][ny][z+1][1-night] = d[x][y][z][night] + 1;
                q.push(make_tuple(nx,ny,z+1,1-night));
            }
        }
        if (d[x][y][z][1-night] == 0) {
            d[x][y][z][1-night] = d[x][y][z][night] + 1;
            q.push(make_tuple(x,y,z,1-night));
        }
    }
    int ans = -1;
    for (int j=0; j<2; j++) {
        for (int i=0; i<=l; i++) {
            if (d[n-1][m-1][i][j] == 0) continue;
            if (ans == -1) {
                ans = d[n-1][m-1][i][j];
            } else if (ans > d[n-1][m-1][i][j]) {
                ans = d[n-1][m-1][i][j];
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 92092 KB | 700 ms | 1695 B |

## C++14

```cpp
#include <iostream>
#include <tuple>
#include <vector>
#include <string>
#include <queue>
using namespace std;
bool check[8][8][9];
int dx[] = {0,0,1,-1,1,-1,1,-1,0};
int dy[] = {1,-1,0,0,1,1,-1,-1,0};
int main() {
    int n = 8;
    vector<string> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    queue<tuple<int,int,int>> q;
    check[7][0][0] = true;
    q.push(make_tuple(7,0,0));
    bool ans = false;
    while (!q.empty()) {
        int x, y, t;
        tie(x,y,t) = q.front(); q.pop();
        if (x == 0 && y == 7) ans = true;
        for (int k=0; k<9; k++) {
            int nx = x+dx[k];
            int ny = y+dy[k];
            int nt = min(t+1, 8);
            if (0 <= nx && nx < n && 0 <= ny && ny < n) {
                if (nx-t >= 0 && a[nx-t][ny] == '#') continue;
                if (nx-t-1 >= 0 && a[nx-t-1][ny] == '#') continue;
                if (check[nx][ny][nt] == false) {
                    check[nx][ny][nt] = true;
                    q.push(make_tuple(nx,ny,nt));
                }
            }
        }
    }
    cout << (ans ? 1 : 0) << '\n';
    return 0;
}
```

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

## C++14

```cpp
#include <iostream>
#include <algorithm>
#include <queue>
#include <tuple>
#include <vector>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
tuple<int,int,int> bfs(vector<vector<int>> &a, int x, int y, int size) {
    int n = a.size();
    vector<tuple<int,int,int>> ans;
    vector<vector<int>> d(n, vector<int> (n, -1));
    queue<pair<int,int>> q;
    q.push(make_pair(x,y));
    d[x][y] = 0;
    while (!q.empty()) {
        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 (0 <= nx && nx < n && 0 <= ny && ny < n && d[nx][ny] == -1) {
                bool ok = false;
                bool eat = false;
                // 아기 상어는 자신의 크기보다 큰 물고기가 있는 칸은 지나갈 수 없고, 나머지 칸은 모두 지나갈 수 있다.
                if (a[nx][ny] == 0) {
                    ok = true;
                } else if (a[nx][ny] < size) { // 아기 상어는 자신의 크기보다 작은 물고기만 먹을 수 있다.
                    ok = eat = true;
                } else if (a[nx][ny] == size) { // 크기가 같은 물고기는 먹을 수 없지만, 그 물고기가 있는 칸은 지나갈 수 있다.
                    ok = true;
                }
                if (ok) {
                    q.push(make_pair(nx,ny));
                    d[nx][ny] = d[x][y] + 1;
                    if (eat) {
                        ans.push_back(make_tuple(d[nx][ny],nx,ny));
                    }
                }
            }
        }
    }
    if (ans.empty()) {
        return make_tuple(-1,-1,-1);
    }
    sort(ans.begin(), ans.end());
    return ans[0];
}
int main() {
    int n;
    cin >> n;
    vector<vector<int>> a(n, vector<int>(n, 0));
    int x, y;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            if (a[i][j] == 9) {
                tie(x, y) = make_pair(i, j);
                a[i][j] = 0;
            }
        }
    }
    int ans = 0;
    int size = 2;
    int exp = 0;
    while (true) {
        int nx, ny, dist;
        tie(dist,nx,ny) = bfs(a, x, y, size);
        if (dist == -1) break;
        a[nx][ny] = 0;
        ans += dist;
        exp += 1;
        if (size == exp) {
            size += 1;
            exp = 0;
        }
        tie(x,y) = make_pair(nx,ny);
    }
    cout << ans << '\n';
    return 0;
}

```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 1992 KB | 4 ms | 2461 B |

## C++14

```cpp
#include <iostream>
#include <queue>
#include <tuple>
#include <string>
#include <algorithm>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int main() {
    int m, n;
    cin >> m >> n;
    vector<string> a(n);
    int sx,sy,ex,ey;
    sx=sy=ex=ey=-1;
    for (int i=0; i<n; i++) {
        cin >> a[i];
        for (int j=0; j<m; j++) {
            if (a[i][j] == 'C') {
                if (sx == -1) {
                    sx = i;
                    sy = j;
                } else {
                    ex = i;
                    ey = j;
                }
            }
        }
    }
    vector<vector<int>> d(n, vector<int>(m, -1));
    queue<pair<int,int>> q;
    d[sx][sy] = 0;
    q.push(make_pair(sx,sy));
    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];
            while (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (a[nx][ny] == '*') break;
                if (d[nx][ny] == -1) {
                    d[nx][ny] = d[x][y] + 1;
                    q.push(make_pair(nx,ny));
                }
                nx += dx[k];
                ny += dy[k];
            }
        }
    }
    cout << d[ex][ey]-1 << '\n';
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
|---|---|---|---|
| 맞았습니다!! | 1992 KB | 0 ms | 1301 B |

## C++14

```cpp
#include <iostream>
#include <cstring>
#include <queue>
#include <string>
#include <algorithm>
using namespace std;
bool prime[10001];
bool c[10001];
int d[10001];
int change(int num, int index, int digit) {
    if (index == 0 && digit == 0) return -1;
    string s = to_string(num);
    s[index] = digit+'0';
    return stoi(s);
}
int main() {
    for (int i=2; i<=10000; i++) {
        if (prime[i] == false) {
            for (int j=i*i; j <= 10000; j+=i) {
                prime[j] = true;
            }
        }
    }
    for (int i=0; i<=10000; i++) {
        prime[i] = !prime[i];
    }
    int t;
    cin >> t;
    while (t--) {
        int n, m;
        cin >> n >> m;
        memset(c,false,sizeof(c));
        memset(d,0,sizeof(d));
        d[n] = 0;
        c[n] = true;
        queue<int> q;
        q.push(n);
        while (!q.empty()) {
            int now = q.front();
            q.pop();
            for (int i=0; i<4; i++) {
                for (int j=0; j<=9; j++) {
                    int next = change(now, i, j);
                    if (next != -1) {
                        if (prime[next] && c[next] == false) {
                            q.push(next);
                            d[next] = d[now] + 1;
                            c[next] = true;
                        }
                    }
                }
            }
        }
        cout << d[m] << '\n';
    }
    return 0;
}
```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 2048 KB | 32 ms | 1418 B |

## C++14

```cpp
#include <iostream>
#include <queue>
#include <vector>
#include <string>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
bool can(bool blind, char from, char to) {
    if (from == to) return true;
    if (blind) {
        if (from == 'R' && to == 'G') return true;
        if (from == 'G' && to == 'R') return true;
    }
    return false;
}
int go(vector<string> &a, bool blind = false) {
    int n = a.size();
    vector<vector<bool>> check(n, vector<bool>(n,false));
    int ans = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (check[i][j] == false) {
                ans += 1;
                queue<pair<int,int>> q;
                q.push(make_pair(i,j));
                check[i][j] = 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 (0 <= nx && nx < n && 0 <= ny && ny < n) {
                            if (check[nx][ny]) continue;
                            if (can(blind,a[x][y],a[nx][ny])) {
                                check[nx][ny] = true;
                                q.push(make_pair(nx,ny));
                            }
                        }
                    }
                }
            }
        }
    }
    return ans;
}
int main() {
    int n;
    cin >> n;
    vector<string> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    cout << go(a) << ' ' << go(a,true) << '\n';
    return 0;
}

```

| 결과 | 메모리 | 시간 | 코드 길이 |
| --- | --- | --- | --- |
| 맞았습니다!! | 1992 KB | 0 ms | 1670 B |

## C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
#include <string>
#include <set>
using namespace std;
const long long limit = 1000000000LL;
int main() {
    long long s, t;
    cin >> s >> t;
    set<long long> check;
    queue<pair<long long,string>> q;
    q.push(make_pair(s,""));
    check.insert(s);
    while (!q.empty()) {
        long long x;
        string str;
        tie(x, str) = q.front(); q.pop();
        if (x == t) {
            if (str.length() == 0) {
                str = "0";
            }
            cout << str << '\n';
            return 0;
        }
        if (0 <= x*x && x*x <= limit && check.count(x*x) == 0) {
            q.push(make_pair(x*x, str+"*"));
            check.insert(x*x);
        }
        if (0 <= x+x && x+x <= limit && check.count(x+x) == 0) {
            q.push(make_pair(x+x, str+"+"));
            check.insert(x+x);
        }
        if (0 <= x-x && x-x <= limit && check.count(x-x) == 0) {
            q.push(make_pair(x-x, str+"-"));
            check.insert(x-x);
        }
        if (x != 0 && 0 <= x/x && x/x <= limit && check.count(x/x) == 0) {
            q.push(make_pair(x/x, str+"/"));
            check.insert(x/x);
        }
    }
    cout << -1 << '\n';
    return 0;
}

```

| 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.