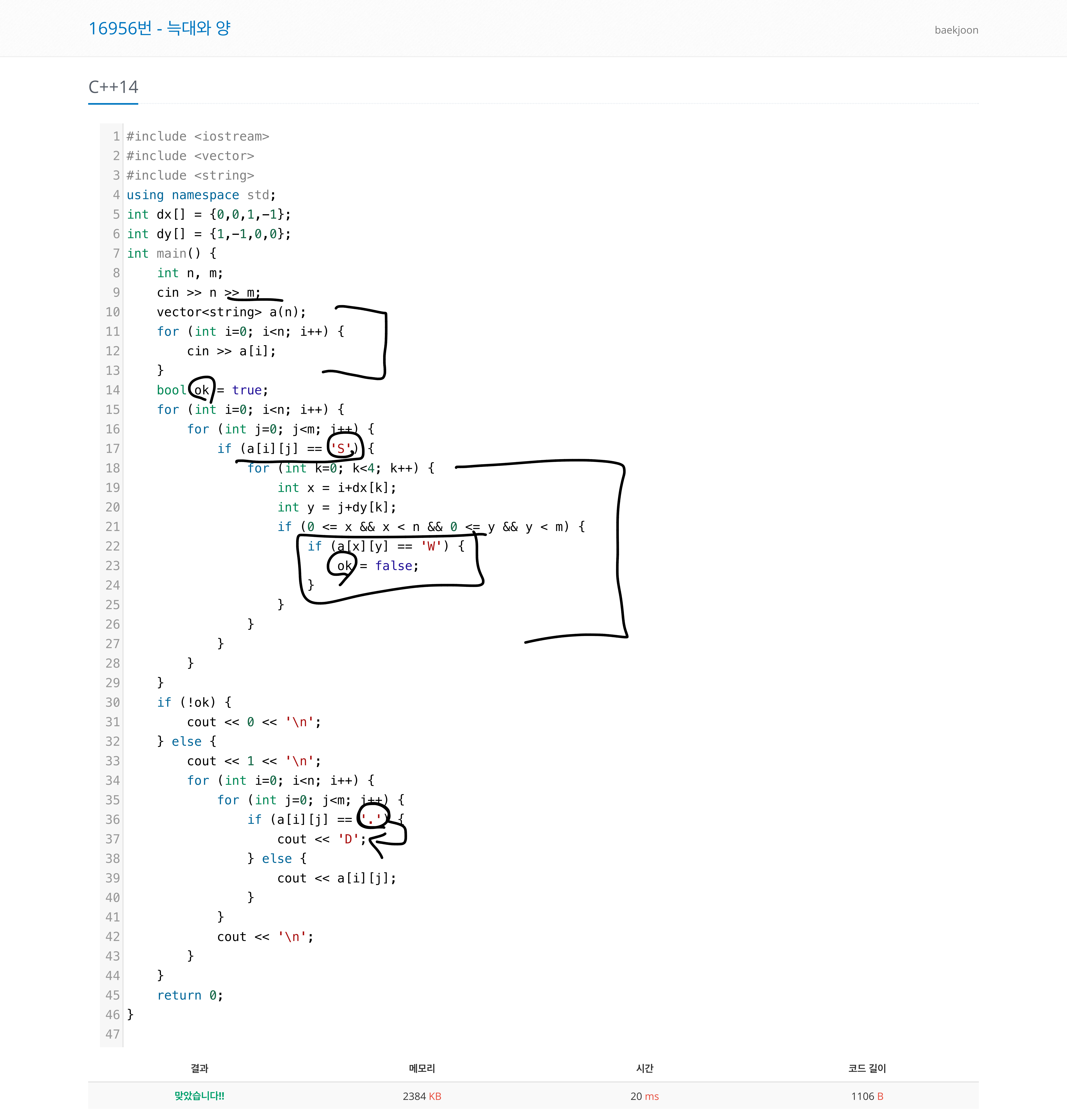
BFS (연습2)

소스코드

최백준 choi@startlink.io



16956번 - 늑대와 양

baekjoon

C++14

```cpp
#include <iostream>
#include <vector>
#include <string>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int main() {
    int n, m;
    cin >> n >> m;
    vector<string> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    bool ok = true;
    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++) {
                    int x = i+dx[k];
                    int y = j+dy[k];
                    if (0 <= x && x < n && 0 <= y && y < m) {
                        if (a[x][y] == 'W') {
                            ok = false;
                        }
                    }
                }
            }
        }
    }
    if (!ok) {
        cout << 0 << '\n';
    } else {
        cout << 1 << '\n';
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j] == '.') {
                    cout << 'D';
                } else {
                    cout << a[i][j];
                }
            }
            cout << '\n';
        }
    }
    return 0;
}

```

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

5014번 - 스타트링크

baekjoon

C++14

```
#include <iostream>
#include <queue>
using namespace std;
int dist[1000001];
bool check[1000001];
int main() {
    int f,s,g,u,d;
    cin >> f >> s >> g >> u >> d;
    queue<int> q;
    q.push(s);
    check[s] = true;
    while (!q.empty()) {
        int now = q.front();
        q.pop();
        if (now + u <= f && check[now+u] == false) {
            dist[now+u] = dist[now] + 1;
            check[now+u] = true;
            q.push(now+u);
        }
        if (now - d >= 1 && check[now-d] == false) {
            dist[now-d] = dist[now] + 1;
            check[now-d] = true;
            q.push(now-d);
        }
    }
    if (check[g]) {
        cout << dist[g] << '\n';
    } else {
        cout << "use the stairs\n";
    }
    return 0;
}
```

거리 (버튼 횟수)

1~F

O(F)

결과	메모리	시간	코드 길이
맞았습니다!!	6872 KB	8 ms	746 B

9376번 - 탈옥

baekjoon

C++14

```cpp
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <deque>
#include <tuple>
using namespace std;
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
vector<vector<int>> bfs(vector<string> &a, int x, int y) {
    int n = a.size();
    int m = a[0].size();
    vector<vector<int>> d(n, vector<int>(m));
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            d[i][j] = -1;
        }
    }
    deque<pair<int,int>> q;
    q.push_back(make_pair(x, y));
    d[x][y] = 0;
    while (!q.empty()) {
        tie(x,y) = q.front(); q.pop_front();
        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 (d[nx][ny] != -1) continue;
            if (a[nx][ny] == '*') continue;
            if (a[nx][ny] == '#') {
                d[nx][ny] = d[x][y] + 1;
                q.push_back(make_pair(nx,ny));
            } else {
                d[nx][ny] = d[x][y];
                q.push_front(make_pair(nx,ny));
            }
        }
    }
    return d;
}
int main() {
    int t;
    cin >> t;
    while (t--) {
        int n, m;
        cin >> n >> m;
        vector<string> a(n+2);
        for (int i=1; i<=n; i++) {
            cin >> a[i];
            a[i] = "." + a[i] + ".";
        }
        n += 2;
        m += 2;
        for (int j=0; j<m; j++) {
            a[0] += ".";
            a[n-1] += ".";
        }
        vector<vector<int>> d0 = bfs(a, 0, 0);
        int x1, y1, x2, y2;
        x1 = y1 = x2 = y2 = -1;
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j] == '$') {
                    if (x1 == -1) {
                        x1 = i;
                        y1 = j;
                    } else {
                        x2 = i;
                        y2 = j;
                    }
                }
            }
        }
        vector<vector<int>> d1 = bfs(a, x1, y1);
        vector<vector<int>> d2 = bfs(a, x2, y2);
        int ans = n*m;
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j] == '*') continue;
                int cur = d0[i][j] + d1[i][j] + d2[i][j];
                if (a[i][j] == '#') cur -= 2;
                if (ans > cur) ans = cur;
            }
        }
        cout << ans << '\n';
    }
    return 0;
}
```

지도

시작

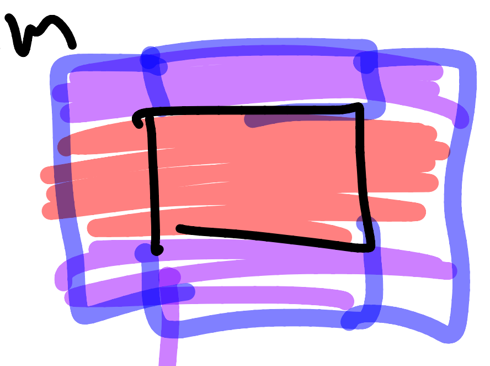
가 0,1

알 되

벽

빈칸

nxm



i, j

결과	메모리	시간	코드 길이
맞았습니다!!	2136 KB	16 ms	2410 B

2251번 - 물통

baekjoon

C++14

A, B, C

0	1	2

```cpp
#include <iostream>
#include <queue>
using namespace std;
bool ans[201];
bool check[201][201];
int cap[3];
int from[] = {0, 0, 1, 1, 2, 2};
int to[] = {1, 2, 0, 2, 0, 1};
int main() {
    for (int i=0; i<3; i++) {
        cin >> cap[i];
    }
    int sum = cap[2];
    queue<pair<int,int>> q;
    q.push(make_pair(0, 0));
    check[0][0] = true;
    ans[cap[2]] = true;
    while (!q.empty()) {
        int cur[3];
        cur[0] = q.front().first;
        cur[1] = q.front().second;
        cur[2] = sum - cur[0] - cur[1];
        q.pop();
        for (int k=0; k<6; k++) {
            int next[3] = {cur[0], cur[1], cur[2]};
            next[to[k]] += next[from[k]];
            next[from[k]] = 0;
            if (next[to[k]] >= cap[to[k]]) {
                next[from[k]] = next[to[k]] - cap[to[k]];
                next[to[k]] = cap[to[k]];
            }
            if (!check[next[0]][next[1]]) {
                check[next[0]][next[1]] = true;
                q.push(make_pair(next[0], next[1]));
                if (next[0] == 0) {
                    ans[next[2]] = true;
                }
            }
        }
    }
    for (int i=0; i<=cap[2]; i++) {
        if (ans[i]) {
            cout << i << ' ';
        }
    }
    cout << '\n';
    return 0;
}
```

ans[i] = T / F : A가 비어 있을 때 C에 물이 i 만큼 들어가는 것이 가능 / 불

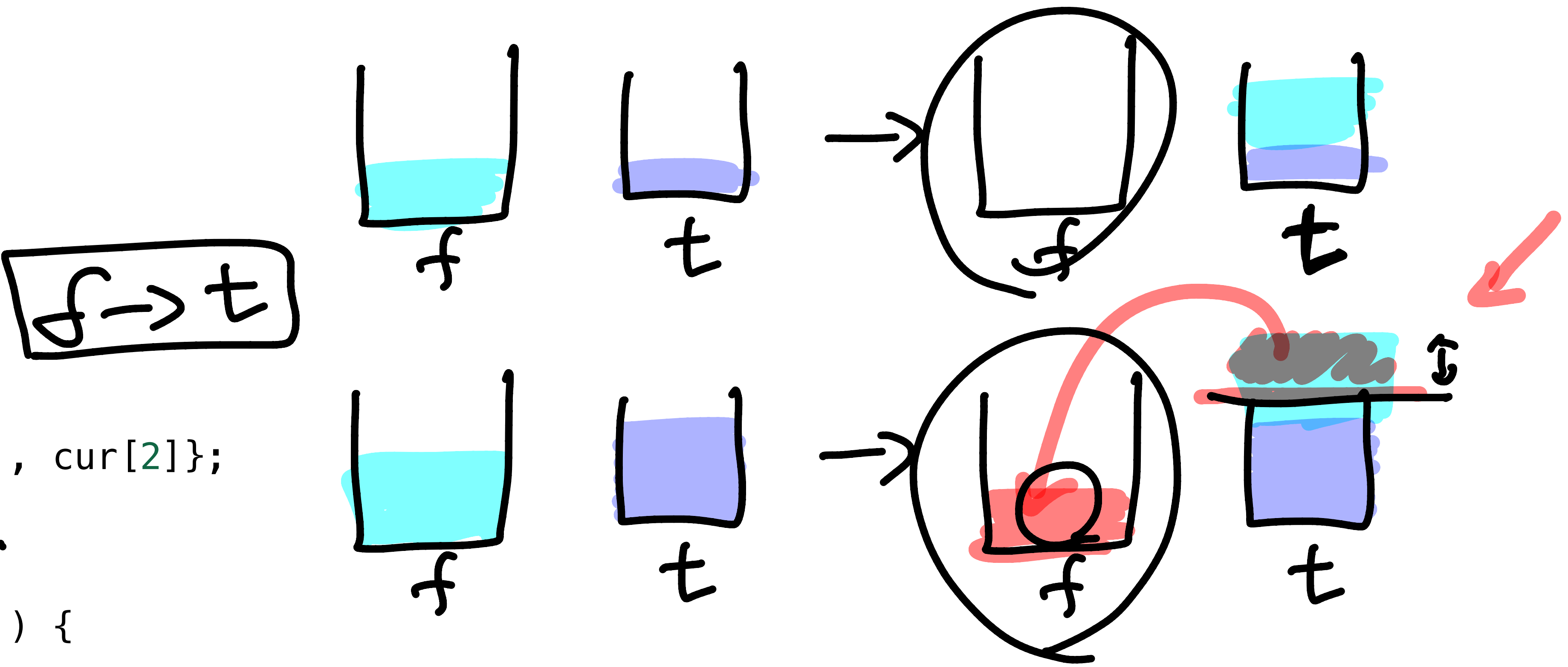
물통의 용량

6가지

f -> t

0 1

(0, 0)



결과	메모리	시간	코드 길이
맞았습니다!!	2028 KB	0 ms	1266 B

16932번 - 모양 만들기

C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
#include <algorithm>
using namespace std;
int n, m;
int a[1000][1000];
int group[1000][1000];
int group_size[1000*1000];
int groupn = 0;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
void bfs(int sx, int sy) {
    groupn += 1;
    queue<pair<int,int>> q;
    q.push(make_pair(sx,sy));
    group[sx][sy] = groupn;
    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 (group[nx][ny] == 0 && a[nx][ny] == 1) {
                    group[nx][ny] = groupn;
                    q.push(make_pair(nx,ny));
                    cnt += 1;
                }
            }
        }
    }
    group_size[groupn] = cnt;
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 1 && group[i][j] == 0) {
                bfs(i, j);
            }
        }
    }
    int ans = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0) {
                vector<int> near;
                for (int k=0; k<4; k++) {
                    int nx = i+dx[k];
                    int ny = j+dy[k];
                    if (0 <= nx && nx < n && 0 <= ny && ny < m) {
                        if (a[nx][ny] == 1) {
                            near.push_back(group[nx][ny]);
                        }
                    }
                }
                sort(near.begin(), near.end());
                near.erase(unique(near.begin(), near.end()), near.end());
                int sum = 1;
                for (int neighbor : near) {
                    sum += group_size[neighbor];
                }
                if (ans < sum) ans = sum;
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

그룹 = 모양

그룹의 크기

그룹의 개수

① 연결

인접한

그룹 1

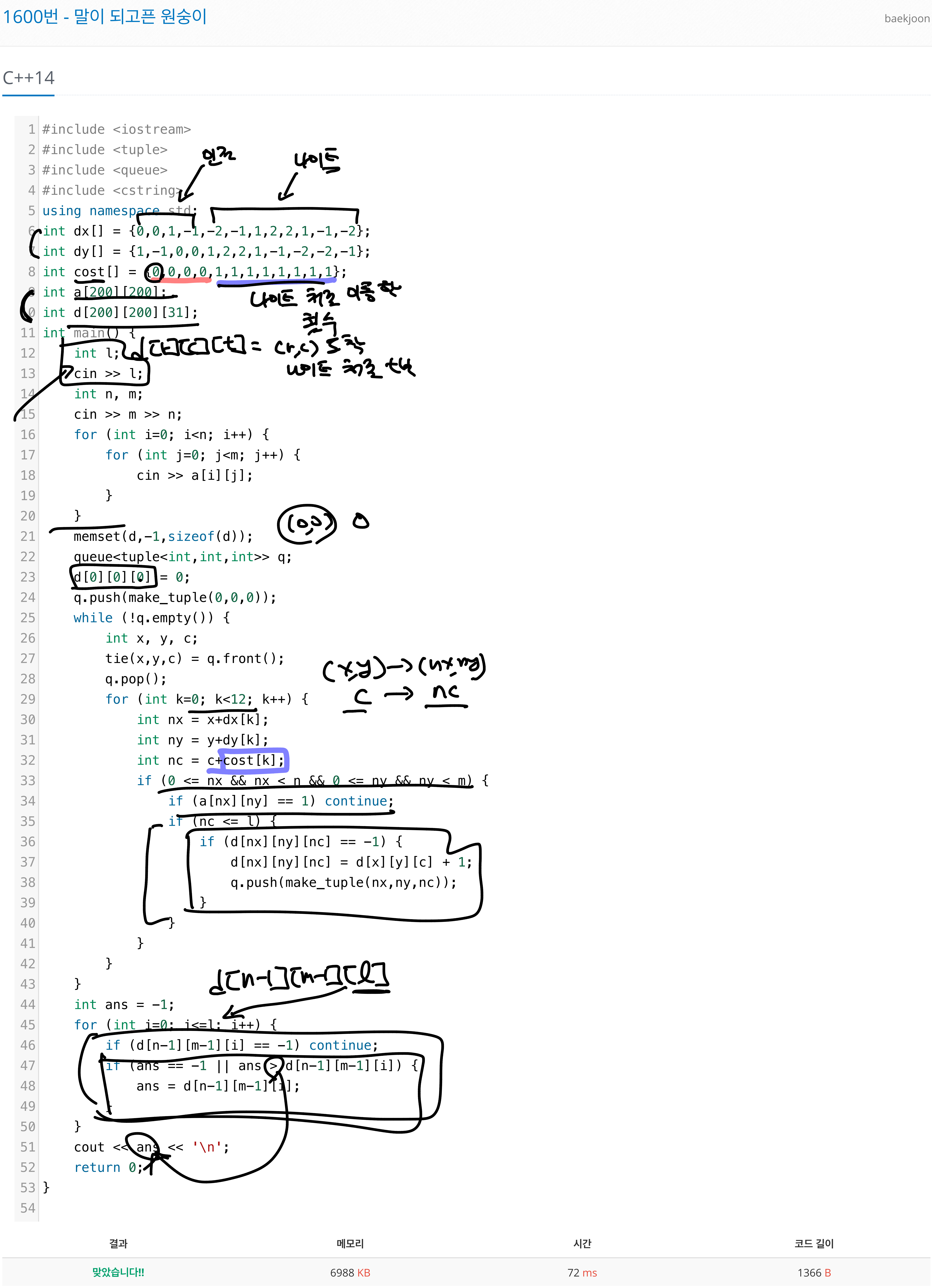
0 —> 1

중복제거

0 —> 1

O(NM)

결과	메모리	시간	코드 길이
맞았습니다!!	13708 KB	240 ms	2068 B



1600번 - 말이 되고픈 원숭이

baekjoon

C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
#include <cstring>
using namespace std;
int dx[] = {0,0,1,-1,-2,-1,1,2,2,1,-1,-2};
int dy[] = {1,-1,0,0,1,2,2,1,-1,-2,-2,-1};
int cost[] = {0,0,0,0,1,1,1,1,1,1,1,1};
int a[200][200];
int d[200][200][31];
int main() {
    int l;
    cin >> l;
    int n, m;
    cin >> m >> n;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    memset(d,-1,sizeof(d));
    queue<tuple<int,int,int>> q;
    d[0][0][0] = 0;
    q.push(make_tuple(0,0,0));
    while (!q.empty()) {
        int x, y, c;
        tie(x,y,c) = q.front();
        q.pop();
        for (int k=0; k<12; k++) {
            int nx = x+dx[k];
            int ny = y+dy[k];
            int nc = c+cost[k];
            if (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (a[nx][ny] == 1) continue;
                if (nc <= l) {
                    if (d[nx][ny][nc] == -1) {
                        d[nx][ny][nc] = d[x][y][c] + 1;
                        q.push(make_tuple(nx,ny,nc));
                    }
                }
            }
        }
    }
    int ans = -1;
    for (int i=0; i<=l; i++) {
        if (d[n-1][m-1][i] == -1) continue;
        if (ans == -1 || 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!!	6988 KB	72 ms	1366 B

17086번 - 아기 상어 2

baekjoon

C++14

```cpp
#include <iostream>
#include <tuple>
#include <queue>
using namespace std;
int a[55][55];
int d[55][55];
int n, m;
int dx[] = {0,0,1,-1,1,1,-1,-1};
int dy[] = {1,-1,0,0,1,-1,1,-1};
int go(int sx, int sy) {
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            d[i][j] = -1;
        }
    }
    d[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<8; k++) {
            int nx = x+dx[k];
            int ny = y+dy[k];
            if (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (d[nx][ny] == -1) {
                    if (a[nx][ny] == 1) {
                        return d[x][y]+1;
                    } else {
                        q.push(make_pair(nx,ny));
                        d[nx][ny] = d[x][y] + 1;
                    }
                }
            }
        }
    }
    return 0;
}
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 i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0) {
                int dist = go(i, j);
                if (ans < dist) ans = dist;
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

50×50

8개의 방향

(sx, sy) 칸과 모든 칸으로의 거리

(안전 거리)

0 → 1 → 2

이동 X

상어

빈칸

NM

NM

$O((NM)^2)$

결과	메모리	시간	코드 길이
맞았습니다!!	2012 KB	80 ms	1366 B

4991번 - 로봇 청소기

baekjoon

C++14

BFS

(sx,sy) 시작 부분

```cpp
#include <iostream>
#include <algorithm>
#include <tuple>
#include <queue>
#include <string>
#include <vector>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
vector<vector<int>> bfs(vector<string> &a, int sx, int sy) {
    int n = a.size();
    int m = a[0].size();
    vector<vector<int>> dist(n, vector<int>(m,-1));
    queue<pair<int,int>> q;
    q.push(make_pair(sx,sy));
    dist[sx][sy] = 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 (0 <= nx && nx < n && 0 <= ny && ny < m) {
                if (dist[nx][ny] == -1 && a[nx][ny] != 'x') {
                    dist[nx][ny] = dist[x][y] + 1;
                    q.push(make_pair(nx,ny));
                }
            }
        }
    }
    return dist;
}
int main() {
    while (true) {
        int n, m;
        cin >> m >> n;
        if (n == 0 && m == 0) break;
        vector<string> a(n);
        for (int i=0; i<n; i++) {
            cin >> a[i];
        }
        vector<pair<int,int>> b(1);
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j] == 'o') {
                    b[0] = make_pair(i,j);
                } else if (a[i][j] == '*') {
                    b.push_back(make_pair(i,j));
                }
            }
        }
        int l = b.size();
        vector<vector<int>> d(l, vector<int>(l));
        bool ok = true;
        for (int i=0; i<l; i++) {
            auto dist = bfs(a,b[i].first,b[i].second);
            for (int j=0; j<l; j++) {
                d[i][j] = dist[b[j].first][b[j].second];
                if (d[i][j] == -1) {
                    ok = false;
                }
            }
        }
        if (ok == false) {
            cout << -1 << '\n';
            continue;
        }
        vector<int> p(l-1);
        for (int i=0; i<l-1; i++) {
            p[i] = i+1;
        }
        int ans = -1;
        do {
            int now = d[0][p[0]];
            for (int i=0; i<l-2; i++) {
                now += d[p[i]][p[i+1]];
            }
            if (ans == -1 || ans > now) {
                ans = now;
            }
        } while(next_permutation(p.begin(), p.end()));
        cout << ans << '\n';
    }
    return 0;
}
```

(0,0) →

b[i] =

i번 더러운 칸의 위치

d[i][j] = i번 → j번 거리

10! x 10

BFS 결과 이용

10! x NM

결과	메모리	시간	코드 길이
맞았습니다!!	1996 KB	52 ms	2346 B

2151번 - 거울 설치

baekjoon

C++14

```cpp
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int main() {
    int n;
    cin >> n;
    vector<string> s(n);
    vector<vector<int>> b(n, vector<int>(n));
    vector<pair<int,int>> v;
    int start=-1, end=-1;
    for (int i=0; i<n; i++) {
        cin >> s[i];
        for (int j=0; j<n; j++) {
            if (s[i][j] == '#') {
                if (start == -1) {
                    start = v.size();
                } else {
                    end = v.size();
                }
                v.push_back(make_pair(i,j));
                b[i][j] = v.size()-1;
            } else if (s[i][j] == '!') {
                v.push_back(make_pair(i,j));
                b[i][j] = v.size()-1;
            }
        }
    }
    int m = v.size();
    vector<vector<bool>> a(m, vector<bool>(m, false));
    for (int i=0; i<v.size(); i++) {
        for (int k=0; k<4; k++) {
            int x = v[i].first+dx[k];
            int y = v[i].second+dy[k];
            while (0 <= x && x < n && 0 <= y && y < n) {
                if (s[x][y] == '*') break;
                if (s[x][y] == '!' || s[x][y] == '#') {
                    a[i][b[x][y]] = true;
                }
                x += dx[k];
                y += dy[k];
            }
        }
    }
    queue<int> q;
    vector<int> dist(m, -1);
    q.push(start);
    dist[start] = 0;
    while (!q.empty()) {
        int now = q.front();
        q.pop();
        for (int i=0; i<m; i++) {
            if (a[now][i] && dist[i] == -1) {
                dist[i] = dist[now]+1;
                q.push(i);
            }
        }
    }
    cout << dist[end]-1 << '\n';
    return 0;
}
```

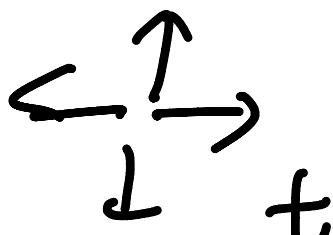
4

b[i][j]번 거울
거울의 번호

v[i]=
i번 거울의
위치

a[i][j] = i → j

j번 거울이
i번 거울에



the [illegible]

결과	메모리	시간	코드 길이
맞았습니다!!	2128 KB	0 ms	1704 B

2234번 - 성곽

baekjoon

C++14

방 번호

2^0 2^1 ↑ 2 4 2^2 ← 1 → ↓ 8 2^3

성곽

d[i][j]: (i,j) 방번호

room[i]: 방 i의 크기

방번호

(x,y) 시작

$2^k = (1<<k)$

벽이

NM

NM

O(NM)

```cpp
#include <iostream>
#include <queue>
using namespace std;
int n, m;
int a[50][50];
int d[50][50];
int room[50*50];
int dx[] = {0,-1,0,1};
int dy[] = {-1,0,1,0};
int bfs(int x, int y, int rooms) {
    queue<pair<int,int>> q;
    q.push(make_pair(x,y));
    d[x][y] = rooms;
    int cnt = 0;
    while (!q.empty()) {
        x = q.front().first;
        y = q.front().second;
        q.pop();
        cnt += 1;
        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 (d[nx][ny] != 0) continue;
            if (a[x][y] & (1<<k)) continue;
            q.push(make_pair(nx,ny));
            d[nx][ny] = rooms;
        }
    }
    return cnt;
}
int main() {
    cin >> m >> n;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    int rooms = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            if (d[i][j] == 0) {
                rooms += 1;
                room[rooms] = bfs(i, j, rooms);
            }
        }
    }
    cout << rooms << '\n';
    int ans = 0;
    for (int i=1; i<=rooms; i++) {
        if (ans < room[i]) {
            ans = room[i];
        }
    }
    cout << ans << '\n';
    ans = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<m; j++) {
            int x = i;
            int y = j;
            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 (d[nx][ny] == d[x][y]) continue;
                if (a[x][y] & (1<<k)) {
                    if (ans < room[d[x][y]]+room[d[nx][ny]]) {
                        ans = room[d[x][y]]+room[d[nx][ny]];
                    }
                }
            }
        }
    }
    cout << ans << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	2016 KB	0 ms	1921 B

12906번 - 새로운 하노이 탑

baekjoon

C++14

```cpp
#include <iostream>
#include <map>
#include <queue>
#include <array>
using namespace std;
int main() {
    array<string,3> s;
    for (int i=0; i<3; i++) {
        int cnt;
        cin >> cnt;
        if (cnt > 0) {
            cin >> s[i];
        } else {
            s[i] = "";
        }
    }
    int cnt[3] = {0, 0, 0};
    for (int i=0; i<3; i++) {
        for (int j=0; j<s[i].length(); j++) {
            cnt[s[i][j]-'A'] += 1;
        }
    }
    map<array<string,3>, int> d;
    queue<array<string,3>> q;
    q.push(s);
    d[s] = 0;
    while (!q.empty()) {
        auto now = q.front();
        q.pop();
        for (int i=0; i<3; i++) {
            for (int j=0; j<3; j++) {
                if (i == j) continue;
                if (now[i].length() == 0) continue;
                array<string,3> next(now);
                next[j].push_back(next[i].back());
                next[i].pop_back();
                if (d.count(next) == 0) {
                    d[next] = d[now] + 1;
                    q.push(next);
                }
            }
        }
    }
    array<string,3> ans;
    for (int i=0; i<3; i++) {
        for (int j=0; j<cnt[i]; j++) {
            ans[i] += (char)('A' + i);
        }
    }
    cout << d[ans] << '\n';
    return 0;
}
```

탑: ~~A B C~~
0 1 2

S[i] = i번 탑의 원판

cnt[i]: i번 원판의 개수

d[" "][" "][" "]

6가지
A->B
A->C
B->A
B->C
C->A
C->B

i->j

B C

결과	메모리	시간	코드 길이
맞았습니다!!	45592 KB	1424 ms	1266 B

17141번 - 연구소 2

C++14

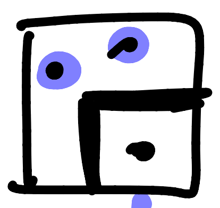
```cpp
#include <iostream>
#include <tuple>
#include <queue>
#include <cstring>
using namespace std;
int a[100][100];
int d[100][100];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int n, m;
vector<pair<int,int>> candi;
int ans = -1;
void bfs() {
    memset(d,-1,sizeof(d));
    queue<pair<int,int>> q;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j] == 3) {
                q.push(make_pair(i,j));
                d[i][j] = 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 (0 <= nx && nx < n && 0 <= ny && ny < n) {
                if (a[nx][ny] != 1 && d[nx][ny] == -1) {
                    d[nx][ny] = d[x][y] + 1;
                    q.push(make_pair(nx,ny));
                }
            }
        }
    }
    int cur = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j] != 1) {
                if (d[i][j] == -1) return;
                if (cur < d[i][j]) cur = d[i][j];
            }
        }
    }
    if (ans == -1 || ans > cur) {
        ans = cur;
    }
}
void go(int index, int cnt) {
    if (index == candi.size()) {
        if (cnt == m) {
            bfs();
        }
    } else {
        int x, y;
        tie(x,y) = candi[index];
        a[x][y] = 3;
        go(index+1, cnt+1);
        a[x][y] = 0;
        go(index+1, cnt);
    }
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            if (a[i][j] == 2) {
                a[i][j] = 0;
                candi.push_back(make_pair(i,j));
            }
        }
    }
    go(0,0);
    cout << ans << '\n';
    return 0;
}
```

-1

시작위치

벽이 아니라

방문



2개

2중

실제 바이러스

입력

0 : 빈칸
1 : 벽
2 : 바이러스

최대 10

결과	메모리	시간	코드 길이
맞았습니다!!	2064 KB	16 ms	1820 B

17142번 - 연구소 3

C++14

```
#include <iostream>
#include <tuple>
#include <queue>
#include <cstring>
using namespace std;
int a[100][100];
int d[100][100];
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
int n, m;
vector<pair<int,int>> candi;
int ans = -1;
void bfs() {
    memset(d,-1,sizeof(d));
    queue<pair<int,int>> q;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j] == 3) {
                q.push(make_pair(i,j));
                d[i][j] = 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 (0 <= nx && nx < n && 0 <= ny && ny < n) {
                if (a[nx][ny] != 1 && d[nx][ny] == -1) {
                    d[nx][ny] = d[x][y] + 1;
                    q.push(make_pair(nx,ny));
                }
            }
        }
    }
    int cur = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j] == 0) { // 차이 3
                if (d[i][j] == -1) return;
                if (cur < d[i][j]) cur = d[i][j];
            }
        }
    }
    if (ans == -1 || ans > cur) {
        ans = cur;
    }
}
void go(int index, int cnt) {
    if (index == candi.size()) {
        if (cnt == m) {
            bfs();
        }
    } else {
        int x, y;
        tie(x,y) = candi[index];
        a[x][y] = 3;
        go(index+1, cnt+1);
        a[x][y] = 2; // 차이 2
        go(index+1, cnt);
    }
}
int main() {
    cin >> n >> m;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> a[i][j];
            if (a[i][j] == 2) {
                // 차이 1 (연구소 2는 a[i][j] = 0; 이 적혀있음)
                candi.push_back(make_pair(i,j));
            }
        }
    }
    go(0,0);
    cout << ans << '\n';
    return 0;
}
```

ㅐ O

a[i][j] != 1

활성

a[x][y] = 0

비활성

결과	메모리	시간	코드 길이
맞았습니다!!	2064 KB	16 ms	1890 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.