최백준 choi@startlink.io

C++14

```cpp
#include <iostream>
#include <queue>
#include <string>
#include <map>
using namespace std;
int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};
int main() {
    int n = 3;
    int start = 0;
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            int temp;
            cin >> temp;
            if (temp == 0) {
                temp = 9;
            }
            start = start * 10 + temp;
        }
    }
    queue<int> q;
    map<int,int> dist;
    dist[start] = 0;
    q.push(start);
    while (!q.empty()) {
        int now_num = q.front();
        string now = to_string(now_num);
        q.pop();
        int z = now.find('9');
        int x = z/3;
        int y = z%3;
        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 < n) {
                string next = now;
                swap(next[x*3+y], next[nx*3+ny]);
                int num = stoi(next);
                if (dist.count(num) == 0) {
                    dist[num] = dist[now_num] + 1;
                    q.push(num);
                }
            }
        }
    }
    if (dist.count(123456789) == 0) {
        cout << -1 << '\n';
    } else {
        cout << dist[123456789] << '\n';
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	10440 KB	200 ms	1287 B

Java

```java
import java.util.*;

public class Main {
    public static final int[] dx = {0, 0, 1, -1};
    public static final int[] dy = {1, -1, 0, 0};
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int n = 3;
        int start = 0;
        for (int i=0; i<n; i++) {
            for (int j=0; j<n; j++) {
                int temp = sc.nextInt();
                if (temp == 0) {
                    temp = 9;
                }
                start = start * 10 + temp;
            }
        }
        Queue<Integer> q = new LinkedList<Integer>();
        HashMap<Integer,Integer> d = new HashMap<Integer,Integer>();
        d.put(start, 0);
        q.add(start);
        while (!q.isEmpty()) {
            int now_num = q.remove();
            String now = Integer.toString(now_num);
            int z = now.indexOf('9');
            int x = z/3;
            int y = z%3;
            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 < n) {
                    StringBuilder next = new StringBuilder(now);
                    char temp = next.charAt(x*3+y);
                    next.setCharAt(x*3+y, next.charAt(nx*3+ny));
                    next.setCharAt(nx*3+ny, temp);
                    int num = Integer.parseInt(next.toString());
                    if (!d.containsKey(num)) {
                        d.put(num, d.get(now_num)+1);
                        q.add(num);
                    }
                }
            }
        }
        if (d.containsKey(123456789)) {
            System.out.println(d.get(123456789));
        } else {
            System.out.println("-1");
        }
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	77964 KB	808 ms	1742 B

C++14

```cpp
#include <iostream>
#include <queue>
using namespace std;
bool ans[201];
bool check[201][201];
int cap[3];
int from[3] = {0, 0, 1, 1, 2, 2};
int to[3] = {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;
}
```

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

C++11

```cpp
#include <iostream>
#include <queue>
using namespace std;
bool ans[201];
bool check[201][201];
int main() {
    int a, b, c;
    cin >> a >> b >> c;
    int sum = c;
    queue<pair<int,int>> q;
    q.push(make_pair(0, 0));
    check[0][0] = true;
    ans[c] = true;
    while (!q.empty()) {
        int x = q.front().first;
        int y = q.front().second;
        int z = sum - x - y;
        q.pop();
        int nx, ny, nz;
        nx = x;
        ny = y;
        nz = z;
        // x -> y
        ny += nx;
        nx = 0;
        if (ny >= b) {
            nx = ny-b;
            ny = b;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }

        nx = x;
        ny = y;
        nz = z;
        // x -> z
        nz += nx;
        nx = 0;
        if (nz >= c) {
            nx = nz-c;
            nz = c;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }

        nx = x;
        ny = y;
        nz = z;
        // y -> x
        nx += ny;
        ny = 0;
        if (nx >= a) {
            ny = nx-a;
            nx = a;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }

        nx = x;
        ny = y;
        nz = z;
        // y -> z
        nz += ny;
        ny = 0;
        if (nz >= c) {
            ny = nz-c;
            nz = c;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }

        nx = x;
        ny = y;
        nz = z;
        // z -> x
        nx += nz;
        nz = 0;
        if (nx >= a) {
            nz = nx-a;
            nx = a;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }

        nx = x;
        ny = y;
        nz = z;
        // z -> y
        ny += nz;
        nz = 0;
        if (ny >= b) {
            nz = ny-b;
            ny = b;
        }
        if (!check[nx][ny]) {
            check[nx][ny] = true;
            q.push(make_pair(nx,ny));
            if (nx == 0) {
                ans[nz] = true;
            }
        }
    }
    for (int i=0; i<=c; i++) {
        if (ans[i]) {
            cout << i << ' ';
        }
    }
    cout << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!	2028 KB	0 ms	2736 B

Java

```java
import java.util.*;

class Pair implements Comparable<Pair> {
    final int first;
    final int second;

    Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }

    @Override
    public int compareTo(Pair pair) {
        if (this.first < pair.first) {
            return -1;
        }
        if (this.first > pair.first) {
            return 1;
        }
        if (this.second < pair.second) {
            return -1;
        }
        if (this.second > pair.second) {
            return 1;
        }
        return 0;
    }

    public boolean equals(Object object) {
        if (object instanceof Pair) {
            Pair pair = (Pair)object;
            return this.first == pair.first && this.second == pair.second;
        }
        return false;
    }

    public int hashCode() {
        int n = 1;
        n = 19 * n + this.first;
        n = 19 * n + this.second;
        return n;
    }
}
public class Main {
    public static final int[] from = {0, 0, 1, 1, 2, 2};
    public static final int[] to = {1, 2, 0, 2, 0, 1};
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int[] cap = new int[3];
        for (int i=0; i<3; i++) {
            cap[i] = sc.nextInt();
        }
        int sum = cap[2];
        boolean[][] check = new boolean[201][201];
        boolean[] ans = new boolean[201];
        Queue<Pair> q = new LinkedList<Pair>();
        q.add(new Pair(0, 0));
        check[0][0] = true;
        ans[cap[2]] = true;
        while (!q.isEmpty()) {
            int[] cur = new int[3];
            Pair p = q.peek();
            cur[0] = p.first;
            cur[1] = p.second;
            cur[2] = sum - cur[0] - cur[1];
            q.remove();
            for (int k=0; k<6; k++) {
                int[] next = {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.add(new Pair(next[0], next[1]));
                    if (next[0] == 0) {
                        ans[next[2]] = true;
                    }
                }
            }
        }
        for (int i=0; i<=cap[2]; i++) {
            if (ans[i]) {
                System.out.print(i + " ");
            }
        }
        System.out.println();
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!	11592 KB	112 ms	2642 B

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;
}
```

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

Java

```java
import java.util.*;
class Pair {
    int x, y;
    Pair(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
public class Main {
    public static int[] dx = {1, -1, 0, 0};
    public static int[] dy = {0, 0, 1, -1};
    public static int[][] bfs(String[] a, int x, int y) {
        int n = a.length;
        int m = a[0].length();
        int[][] d = new int[n][m];
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                d[i][j] = -1;
            }
        }
        ArrayDeque<Pair> deque = new ArrayDeque<Pair>();
        deque.add(new Pair(x, y));
        d[x][y] = 0;
        while (!deque.isEmpty()) {
            Pair p = deque.poll();
            x = p.x;
            y = p.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 (d[nx][ny] != -1) continue;
                char c = a[nx].charAt(ny);
                if (c == '*') continue;
                if (c == '#') {
                    d[nx][ny] = d[x][y] + 1;
                    deque.addLast(new Pair(nx, ny));
                } else {
                    d[nx][ny] = d[x][y];
                    deque.addFirst(new Pair(nx, ny));
                }
            }
        }
        return d;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while (t-- > 0) {
            int n = sc.nextInt();
            int m = sc.nextInt();
            sc.nextLine();
            String[] a = new String[n+2];
            for (int i=1; i<=n; i++) {
                a[i] = sc.nextLine();
                a[i] = "." + a[i] + ".";
            }
            n += 2;
            m += 2;
            a[0] = a[n-1] = "";
            for (int j=0; j<m; j++) {
                a[0] += ".";
                a[n-1] += ".";
            }
            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].charAt(j) == '$') {
                        if (x1 == -1) {
                            x1 = i;
                            y1 = j;
                        } else {
                            x2 = i;
                            y2 = j;
                        }
                    }
                }
            }
            int[][] d1 = bfs(a, x1, y1);
            int[][] d2 = bfs(a, x2, y2);
            int ans = n*m;
            for (int i=0; i<n; i++) {
                for (int j=0; j<m; j++) {
                    char c = a[i].charAt(j);
                    if (c == '*') continue;
                    int cur = d0[i][j] + d1[i][j] + d2[i][j];
                    if (c == '#') cur -= 2;
                    if (ans > cur) ans = cur;
                }
            }
            System.out.println(ans);
        }
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	31660 KB	324 ms	2999 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

Java

```java
import java.util.*;
public class Main {
    static char[][] a = new char[111][111];
    static boolean[][] c = new boolean[111][111];
    static boolean[] key = new boolean[111];
    static int[] dx = {0,0,1,-1};
    static int[] dy = {1,-1,0,0};
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while (t-- > 0) {
            int n = sc.nextInt();
            int m = sc.nextInt();
            for (int i=0; i<111; i++) {
                for (int j=0; j<111; j++) {
                    a[i][j] = 0;
                }
            }
            for (int i=2; i<n+2; i++) {
                String line = sc.next();
                for (int j=0; j<m; j++) {
                    a[i][j+2] = line.charAt(j);
                }
            }
            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] = '*';
            }
            Arrays.fill(key, false);
            String temp = sc.next();
            int len = temp.length();
            if (temp.charAt(0) != '0') {
                for (int i=0; i<len; i++) {
                    key[temp.charAt(i)-'a'] = true;
                }
            }
            int ans = 0;
            for (int i=0; i<111; i++) {
                Arrays.fill(c[i], false);
            }
            Queue<Integer> q = new LinkedList<>();
            Queue<Integer>[] door = new LinkedList[26];
            for (int i=0; i<26; i++) {
                door[i] = new LinkedList<Integer>();
            }
            q.add(1); q.add(1);
            c[1][1] = true;
            while (!q.isEmpty()) {
                int x = q.remove();
                int y = q.remove();
                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.add(nx); q.add(ny);
                    } else if (w == '$') {
                        ans += 1;
                        q.add(nx); q.add(ny);
                    } else if (w >= 'A' && w <= 'Z') {
                        if (key[w-'A']) {
                            q.add(nx); q.add(ny);
                        } else {
                            door[w-'A'].add(nx);
                            door[w-'A'].add(ny);
                        }
                    } else if (w >= 'a' && w <= 'z') {
                        q.add(nx); q.add(ny);
                        if (!key[w-'a']) {
                            key[w-'a'] = true;
                            while (!door[w-'a'].isEmpty()) {
                                q.add(door[w-'a'].remove());
                            }
                        }
                    }
                }
            }
            System.out.println(ans);
        }
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	23244 KB	284 ms	3363 B

C++14

```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;
}
```

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

Java

```java
import java.util.*;
class Pair {
    int first;
    int second;
    Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }
}
public class Main {
    static final int[] dx = {0,0,1,-1};
    static final int[] dy = {1,-1,0,0};
    static int[][] bfs(String[] a, int sx, int sy) {
        int n = a.length;
        int m = a[0].length();
        int[][] dist = new int[n][m];
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                dist[i][j] = -1;
            }
        }
        Queue<Pair> q = new LinkedList<Pair>();
        q.add(new Pair(sx,sy));
        dist[sx][sy] = 0;
        while (!q.isEmpty()) {
            Pair p = q.remove();
            int x = p.first;
            int y = p.second;
            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].charAt(ny) != 'x') {
                        dist[nx][ny] = dist[x][y] + 1;
                        q.add(new Pair(nx,ny));
                    }
                }
            }
        }
        return dist;
    }
static boolean next_permutation(int[] a) {
        int i = a.length-1;
        while (i > 0 && a[i-1] >= a[i]) {
            i -= 1;
        }
        if (i <= 0) {
            return false;
        }
        int j = a.length-1;
        while (a[j] <= a[i-1]) {
            j -= 1;
        }
        int temp = a[i-1];
        a[i-1] = a[j];
        a[j] = temp;
        j = a.length-1;
        while (i < j) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
            i += 1;
            j -= 1;
        }
        return true;
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        while (true) {
            int m = sc.nextInt();
            int n = sc.nextInt();
            if (n == 0 && m == 0) break;
            String[] a = new String[n];
            for (int i=0; i<n; i++) {
                a[i] = sc.next();
            }
            ArrayList<Pair> b = new ArrayList<>();
            b.add(new Pair(0,0));
            for (int i=0; i<n; i++) {
                for (int j=0; j<m; j++) {
                    char x = a[i].charAt(j);
                    if (x == 'o') {
                        b.set(0, new Pair(i,j));
                    } else if (x == '*') {
                        b.add(new Pair(i,j));
                    }
                }
            }
            int l = b.size();
            int[][] d = new int[l][l];
            boolean ok = true;
            for (int i=0; i<l; i++) {
                int[][] dist = bfs(a, b.get(i).first, b.get(i).second);
                for (int j=0; j<l; j++) {
                    d[i][j] = dist[b.get(j).first][b.get(j).second];
                    if (d[i][j] == -1) {
                        ok = false;
                    }
                }
            }
            if (ok == false) {
                System.out.println(-1);
                continue;
            }
            int[] p = new int[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));
            System.out.println(ans);
        }
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	27116 KB	640 ms	3614 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

Java

```java
import java.util.*;
class Pair {
    int first;
    int second;
    Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }
}
public class Main {
    static final int[] dx = {0,0,1,-1};
    static final int[] dy = {1,-1,0,0};
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int m = sc.nextInt();
        int n = sc.nextInt();
        String[] a = new String[n];
        int sx,sy,ex,ey;
        sx=sy=ex=ey=-1;
        for (int i=0; i<n; i++) {
            a[i] = sc.next();
            for (int j=0; j<m; j++) {
                if (a[i].charAt(j) == 'C') {
                    if (sx == -1) {
                        sx = i;
                        sy = j;
                    } else {
                        ex = i;
                        ey = j;
                    }
                }
            }
        }
        int[][] d = new int[n][m];
        for (int i=0; i<n; i++) {
            for (int j=0; j<m; j++) {
                d[i][j] = -1;
            }
        }
        Queue<Pair> q = new LinkedList<Pair>();
        q.add(new Pair(sx,sy));
        d[sx][sy] = 0;
        while (!q.isEmpty()) {
            Pair p = q.remove();
            int x = p.first;
            int y = p.second;
            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].charAt(ny) == '*') break;
                    if (d[nx][ny] == -1) {
                        d[nx][ny] = d[x][y] + 1;
                        q.add(new Pair(nx,ny));
                    }
                    nx += dx[k];
                    ny += dy[k];
                }
            }
        }
        System.out.println(d[ex][ey]-1);
    }
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	12716 KB	184 ms	1836 B

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

Java

```java
import java.util.*;
public class Main {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
        while (t-- > 0) {
            int n = sc.nextInt();
            int[] from = new int[n];
            int[] how = new int[n];
            int[] dist = new int[n];
            for (int i=0; i<n; i++) {
                from[i] = how[i] = dist[i] = -1;
            }
            Queue<Integer> q = new LinkedList<>();
            q.add(1%n);
            dist[1%n] = 0;
            how[1%n] = 1;
            while (!q.isEmpty()) {
                int now = q.remove();
                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.add(next);
                    }
                }
            }
            if (dist[0] == -1) {
                System.out.println("BRAK");
            } else {
                StringBuilder ans = new StringBuilder();
                for (int i=0; i!=-1; i=from[i]) {
                    ans.append(Integer.toString(how[i]));
                }
                System.out.println(ans.reverse());
            }
        }
    }
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	12760 KB	128 ms	1352 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

Java

```java
import java.util.*;
public class Main {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        String[] a = new String[2];
        for (int i=0; i<2; i++) {
            a[i] = sc.next();
        }
        int[][] dirs = {{0,1},{0,-1},{1,k}};
        int[][] d = new int[2][n];
        for (int i=0; i<2; i++) {
            Arrays.fill(d[i],-1);
        }
        Queue<Integer> q = new LinkedList<>();
        d[0][0] = 0;
        q.add(0); q.add(0);
        boolean ok = false;
        while (!q.isEmpty()) {
            int x = q.remove();
            int y = q.remove();
            for (int[] dir : dirs) {
                int dx = dir[0];
                int dy = dir[1];
                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].charAt(ny) == '0') continue;
                if (ny < d[x][y]+1) continue;
                d[nx][ny] = d[x][y] + 1;
                q.add(nx); q.add(ny);
            }
            if (ok) break;
        }
        System.out.println(ok ? "1" : "0");
    }
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	31404 KB	372 ms	1321 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.