

# 코딩 테스트 대비 핵심 알고리즘

## 핵심 유형 문제풀이

핵심 유형 문제풀이 | 다양한 문제를 접하며 코딩 테스트에 익숙해지기

강사 나동빈

# 코딩 테스트 대비 핵심 알고리즘

핵심 유형 문제풀이

코딩 테스트 대비  
핵심 유형 문제풀이

혼자 힘으로 풀어보기

코테 대비  
핵심 유형 문제풀이

문제 제목: 연구소3

문제 난이도: ★★★★★

문제 유형: BFS, 완전 탐색

추천 풀이 시간: 60분

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

- $N \times N$  크기의 연구소에 최대 10개의 비활성 바이러스가 존재한다.
- 이 중에서  $M$ 개를 활성 바이러스로 변경할 수 있다.

2	0	0	0	1	1	0
0	0	1	0	1	2	0
0	1	1	0	1	0	0
0	1	0	0	0	0	0
0	0	0	2	0	1	1
0	1	0	0	0	0	0
2	1	0	0	0	0	2

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

- 연구소의 모든 빈 칸에 바이러스가 있게 되는 최소 시간을 출력한다.
- $N = 7, M = 3$ 일 때, 아래와 같이 활성화 시킬 때 시간이 최소가 된다. (벽: -, 빈 칸: 시간)

2	0	0	0	1	1	0
0	0	1	0	1	2	0
0	1	1	0	1	0	0
0	1	0	0	0	0	0
0	0	0	2	0	1	1
0	1	0	0	0	0	0
2	1	0	0	0	0	2



0	1	2	3	-	-	2
1	2	-	3	-	0	1
2	-	-	2	-	1	2
3	-	2	1	2	2	3
3	2	1	0	1	-	-
4	-	2	1	2	3	4
*	-	3	2	3	4	*

## 코딩 테스트 대비 핵심 유형 문제풀이

## 문제 해결 아이디어

## 코테 대비 핵심 유형 문제풀이

- 전체 바이러스 배열에서  **$M$ 개를 선택하는 모든 조합**을 고려해야 한다.
- 이는 다음과 같은 로직에 따라 재귀함수로 구현할 수 있다

$$\text{combination}([1, 2, 3, 4, 5], 2) = \begin{cases} [1] + \text{combination}([2, 3, 4, 5], 1) \\ [2] + \text{combination}([3, 4, 5], 1) \\ [3] + \text{combination}([4, 5], 1) \\ [4] + \text{combination}([5], 1) \end{cases}$$

# 배열에서 N개의 원소를 선택하는 조합(combination)

```
def comb(arr, n):  
    if n == 0:  
        return [[]]  
    result = []  
    for i in range(len(arr)):  
        front = arr[i]  
        for back in comb(arr[i + 1:], n - 1):  
            result.append([front] + back)  
    return result
```



이후에 각 조합마다 *BFS*를 이용해  
바이러스를 퍼뜨린다.

## 코딩 테스트 대비 핵심 유형 문제풀이

## 소스 코드 1)

## 코테 대비

핵심 유형 문제풀이

```
import sys
# 빠른 입력 함수 사용
input = sys.stdin.readline
from collections import deque

# 상, 하, 좌, 우 방향 정의
dx = [-1, 1, 0, 0]
dy = [0, 0, -1, 1]

# 배열에서 N개의 원소를 선택하는 조합(combination)
def comb(arr, n):
    if n == 0:
        return [[]]
    result = []
    for i in range(len(arr)):
        front = arr[i]
        for back in comb(arr[i + 1:], n - 1):
            result.append([front] + back)
    return result
```

```
def bfs(selected): # 너비 우선 탐색(BFS)
    global answer
    # 활성 바이러스(selected)들을 큐에 넣고 방문 처리
    visited = [[-1] * n for _ in range(n)]
    queue = deque()
    for (i, j) in selected:
        visited[i][j] = 0
        queue.append((i, j))
    cnt = 0 # 빈 공간을 바꾼 횟수
    while queue:
        x, y = queue.popleft()
        for i in range(4):
            nx = x + dx[i]
            ny = y + dy[i]
            if nx < 0 or nx >= n or ny < 0 or ny >= n:
                continue
            if visited[nx][ny] == -1 and arr[nx][ny] != 1: # 처음 방문, 벽이 아님
                visited[nx][ny] = visited[x][y] + 1
                queue.append((nx, ny))
                if arr[nx][ny] == 0: # 빈 공간
                    cnt += 1 # 빈 공간을 바꾼 횟수 세기
    taken = 0 # 최대 소요 시간
    for i in range(n):
        for j in range(n):
            # 비활성 바이러스는 처음부터 바이러스
            if arr[i][j] == 0: # 빈 칸에 한해서만 소요 시간 계산
                taken = max(taken, visited[i][j])
    if cnt == target: answer = min(taken, answer)
```

## 코딩 테스트 대비 핵심 유형 문제풀이

## 소스 코드 2)

## 코테 대비.

핵심 유형 문제풀이

```
# 연구소의 크기(N)와 바이러스의 개수(M) 입력
n, m = map(int, input().split())
# 바이러스 위치 정보
viruses = []
# 초기 연구소 맵 정보
arr = []
target = 0 # 빈 칸(퍼뜨려야 할 개수)
answer = int(1e9) # 최종 정답
for i in range(n):
    arr.append(list(map(int, input().split())))
    for j in range(n):
        if arr[i][j] == 2: # 바이러스인 경우
            viruses.append((i, j))
        if arr[i][j] == 0: # 빈 칸인 경우
            target += 1

combinations = comb(viruses, m)
# 모든 활성 바이러스 조합(combination)에 대하여 BFS 수행
for selected in combinations:
    bfs(selected)
if answer == int(1e9): # 바이러스를 모두 퍼뜨릴 수 없음
    print(-1)
else:
    print(answer)
```



코딩 테스트 대비  
핵심 유형 문제풀이

혼자 힘으로 풀어보기

코테 대비  
핵심 유형 문제풀이

문제 제목: 상어 초등학교

문제 난이도: ★★★★★

문제 유형: 시뮬레이션

추천 풀이 시간: 60분

코딩 테스트 대비  
핵심 유형 문제풀이

## 문제 해결 아이디어

코테 대비  
핵심 유형 문제풀이

- 문제의 요구사항 그대로 구현하면 되는 **시뮬레이션** 유형의 문제다.
- 교실의 크기는  $N \times N$ 이고,  $N^2$ 의 각 학생은 자신이 좋아하는 학생이 **4명**씩 있다.
- 이후에 다음의 알고리즘에 따라서 한 명씩 학생을 자리에 배정한다.
  - 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
  - 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
  - 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

코딩 테스트 대비  
핵심 유형 문제풀이

## 문제 해결 아이디어

코테 대비  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

### • 초기 상태


코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 1단계

	4	

코딩 테스트 대비  
핵심 유형 문제풀이

## 문제 해결 아이디어

코테 대비  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

### • 2단계

	3	
	4	

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 3단계

9	3	
	4	

코딩 테스트 대비  
핵심 유형 문제풀이

## 문제 해결 아이디어

코테 대비  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

### • 4단계

9	3	
8	4	

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 5단계

9	3	
8	4	7



코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 6단계

9	3	
8	4	7
		1

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 7단계

9	3	
8	4	7
	6	1

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 8단계

9	3	
8	4	7
5	6	1

코딩 테스트 대비  
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.  
핵심 유형 문제풀이

1. 좋아하는 학생이 인접한 위치에 가장 많은 칸으로 배정
2. 그러한 칸이 여러 개라면 인접한 위치에 비어 있는 칸이 가장 많은 칸으로 배정
3. 그러한 칸도 여러 개라면 행의 번호, 열의 번호가 낮은 순으로 배정

학생의 번호	좋아하는 학생의 번호
4	2, 5, 1, 7
3	1, 9, 4, 5
9	8, 1, 2, 3
8	1, 9, 3, 4
7	2, 3, 4, 8
1	9, 2, 5, 7
6	5, 2, 3, 4
5	1, 9, 2, 8
2	9, 3, 1, 4

• 9단계

9	3	2
8	4	7
5	6	1

## 코딩 테스트 대비 핵심 유형 문제풀이

## 소스 코드 1)

## 코테 대비.

핵심 유형 문제풀이

```
import sys
# 빠른 입력 함수 사용
input = sys.stdin.readline

# 상, 하, 좌, 우 방향 정보
dx = [-1, 1, 0, 0]
dy = [0, 0, -1, 1]

n = int(input()) # 교실의 크기(N)
# 전체 교실(N X N)
arr = [[0] * n for _ in range(n)]
friends_dict = dict()
```

```
def allocate(id, friends):
    max_adj, max_cnt, max_x, max_y = [-1, -1, -1, -1]
    for x in range(n):
        for y in range(n):
            if arr[x][y] != 0: continue # 이미 학생이 있으면 무시
            cnt, adj = [0, 0]
            for i in range(4): # 인접한 칸을 확인하며
                nx = x + dx[i]
                ny = y + dy[i]
                if nx < 0 or ny < 0 or nx >= n or ny >= n:
                    continue
                if arr[nx][ny] == 0: # 비어있는 칸이라면
                    cnt += 1
                if arr[nx][ny] in friends: # 좋아하는 학생이라면
                    adj += 1
            if adj > max_adj: # 좋아하는 학생이 많이 인접한 곳으로
                max_adj, max_cnt, max_x, max_y = adj, cnt, x, y
            elif adj == max_adj: # 여러 칸이라면
                if cnt > max_cnt: # 주변에 비어있는 칸이 많은 곳으로
                    max_cnt, max_x, max_y = [cnt, x, y]
    arr[max_x][max_y] = id # 적절한 위치에 학생 할당
```

## 코딩 테스트 대비 핵심 유형 문제풀이

## 소스 코드 2)

## 코테 대비.

핵심 유형 문제풀이

```
# 한 명씩 학생의 정보를 입력받기
for i in range(n * n):
    row = list(map(int, input().split()))
    id = row[0]
    friends = row[1:]
    friends_dict[id] = friends
# 교실에 학생 할당
allocate(id, friends)
```

```
result = 0 # 전체 만족도의 합
for x in range(n):
    for y in range(n):
        number = 0
        for i in range(4):
            nx = x + dx[i]
            ny = y + dy[i]
            if nx < 0 or ny < 0 or nx >= n or ny >= n:
                continue
            if arr[nx][ny] in friends_dict[arr[x][y]]:
                number += 1
        score = 0 # 현재 학생의 만족도
        if number == 1: score = 1
        if number == 2: score = 10
        if number == 3: score = 100
        if number == 4: score = 1000
        result += score
print(result)
```