

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: 마법사 상어와 비바라기

문제 난이도: ★★☆☆☆

문제 유형: 시뮬레이션

추천 풀이 시간: 60분

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- 전체 맵이 $N \times N$ 격자 형태이다.
- N 이 최대 50이므로, 최대 구름 수는 2,500개이다.
- M (최대 100)번의 이동마다 각 구름이 최대 50번 이동할 수 있다.
- 따라서, 각 구름을 매번 이동시킨다면, 최대 연산 횟수는 약 $2500 \times 100 \times 50$ 이다.
- 이때, 특정 방향으로 50번 이동하는 연산은 1번의 연산으로 대체할 수 있다. (나머지 연산)
- 결과적으로 $2500 \times 100 \times 1$ 번 정도의 연산으로 문제를 해결할 수 있다.

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [초기 상태] 처음에 가장 왼쪽 아래 2 X 2 위치에 비구름이 있다.

0	0	1	0	2
2	3	2	1	0
4	3	2	9	0
1	0	2	9	0
8	8	2	1	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 dx = [0, -1, -1, -1, 0, 1, 1, 1]
 dy = [-1, -1, 0, 1, 1, 1, 0, -1]

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [1단계] 모든 구름이 현재 방향에 따라서 이동한다.

0	0	1	0	2
2	3	2	1	0
4	3	2	9	0
1	0	2	9	0
8	8	2	1	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 dx = [0, -1, -1, -1, 0, 1, 1, 1]
 dy = [-1, -1, 0, 1, 1, 1, 0, -1]

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [1단계] 구름이 있는 칸에 비가 1씩 내리고, 구름은 사라진다.

0	0	1	0	2
2	3	2	1	0
4	3	2	9	0
1	0	3	10	0
8	8	3	2	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 dx = [0, -1, -1, -1, 0, 1, 1, 1]
 dy = [-1, -1, 0, 1, 1, 1, 0, -1]

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [1단계] 대각선 방향에 물이 있는 위치의 개수만큼 물의 양이 증가한다.

0	0	1	0	2
2	3	2	1	0
4	3	2	9	0
1	0	7	12	0
8	8	4	3	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 $dx = [0, -1, -1, -1, 0, 1, 1, 1]$
 $dy = [-1, -1, 0, 1, 1, 1, 0, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [1단계] 구름이 없던 칸 중에서 2 이상인 칸에 구름이 생기고 2만큼 감소한다.

0	0	1	0	0
0	1	0	1	0
2	1	0	7	0
1	0	7	12	0
6	6	4	3	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 dx = [0, -1, -1, -1, 0, 1, 1, 1]
 dy = [-1, -1, 0, 1, 1, 1, 0, -1]

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [2단계] 마찬가지로 방식으로 2단계가 끝난 후의 상태는 다음과 같다.

2	1	1	0	0
0	1	0	1	2
5	4	5	5	0
4	5	12	15	0
4	4	2	1	0

[전체 이동]

이동 방향	이동 크기
1번째	3
3번째	4
8번째	1
4번째	8

차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치에 해당
 $dx = [0, -1, -1, -1, 0, 1, 1, 1]$
 $dy = [-1, -1, 0, 1, 1, 1, 0, -1]$

코딩 테스트 대비 핵심 유형 문제풀이

소스 코드 1)

코테 대비.

핵심 유형 문제풀이

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

n, m = map(int, input().split())
# 초기 구름은 맵의 가장 왼쪽 아래에서 4칸을 차지
clouds = [(n - 1, 0), (n - 1, 1), (n - 2, 0), (n - 2, 1)]

# 차례대로 ←, ↖, ↑, ↗, →, ↘, ↓, ↙ 위치
dx = [0, -1, -1, -1, 0, 1, 1, 1]
dy = [-1, -1, 0, 1, 1, 1, 0, -1]

arr = []
# N X N 맵 정보 입력(물의 수)
for i in range(n):
    arr.append(list(map(int, input().split())))
```

```
# 구름을 이동한 뒤에 비가 내리도록 하는 함수
def rain(d, s):
    positions = [] # 물이 증가한 칸들
    for cloud in clouds:
        x, y = cloud # 현재 구름의 위치(x, y)
        # 현재의 방향으로 s번 이동(위치는 순환)
        x = (x + dx[d - 1] * s) % n
        y = (y + dy[d - 1] * s) % n
        arr[x][y] += 1 # 이동한 뒤에 비 내리기
        positions.append((x, y))
    return positions
```

코딩 테스트 대비 핵심 유형 문제풀이

소스 코드 2)

코테 대비.

핵심 유형 문제풀이

```
for _ in range(m): # M번의 연산을 차례대로 확인
    d, s = map(int, input().split()) # 방향(d)과 이동 횟수(s)
    positions = rain(d, s) # 구름 이동 및 비 내리기
    # 비(구름)가 내린 모든 위치에 대하여, 대각선 방향을 확인해 덧셈
    for position in positions:
        x, y = position
        cnt = 0 # 대각선으로 인접한 위치 중에서 물이 있는 개수
        # 대각선 방향(1, 3, 5, 7)만 확인
        for i in range(1, 8, 2):
            nx = x + dx[i]
            ny = y + dy[i]
            # 범위를 벗어나는 경우 무시
            if nx >= n or ny >= n or nx < 0 or ny < 0:
                continue
            # 인접한 위치에 물이 있다면 카운트
            if arr[nx][ny] >= 1:
                cnt += 1
        arr[x][y] += cnt
    positions = set(positions) # 비가 내린 부분에는 새로운 구름 안 생김
    clouds = [] # 새로운 구름 초기화
    for i in range(n):
        for j in range(n):
            # 값이 2 이상이고, 기존에 구름이 없던 위치에 새로운 구름 생성
            if arr[i][j] >= 2 and (i, j) not in positions:
                clouds.append((i, j))
                arr[i][j] -= 2
```

```
# 정답 출력
answer = 0
for row in arr:
    answer += sum(row)
print(answer)
```

코딩 테스트 대비
핵심 유형 문제풀이

혼자 힘으로 풀어보기

코테 대비.
핵심 유형 문제풀이

문제 제목: 마법사 상어와 파이어볼

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.
- $N \times N$ 크기의 맵에 파이어볼 M 개가 존재한다.
- N 이 최대 50이므로, 원소의 개수는 최대 2,500개다.
- K 번 (최대 1,000)번의 이동이 진행된다.
- 각 파이어볼은 질량 m , 속력 s , 방향 d 를 가진다.
- 단순히 시뮬레이션하여 $O(KM)$ 의 시간 복잡도를 갖는 코드를 작성할 수 있다.

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- [초기 상태] 문제에 제시된 첫 번째 예시를 그림으로 표현하면 다음과 같다.

→			←

파이어볼 목록			
위치(r, c)	질량(m)	속력(s)	방향(d)
(1, 1)	5	2	인덱스 2
(1, 4)	7	1	인덱스 6

차례대로 ↑, ↗, →, ↘, ↓, ↙, ←, ↖ 에 해당
 $dx = [-1, -1, 0, 1, 1, 1, 0, -1]$
 $dy = [0, 1, 1, 1, 0, -1, -1, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- 먼저 모든 파이어볼이 자신의 방향(d)으로 속력(s)만큼 이동한다.

		→	
		←	

파이어볼 목록			
위치(r, c)	질량(m)	속력(s)	방향(d)
(1, 3)	5	2	인덱스 2
(1, 3)	7	1	인덱스 6

차례대로 ↑, ↗, →, ↘, ↓, ↙, ←, ↖ 에 해당
 $dx = [-1, -1, 0, 1, 1, 1, 0, -1]$
 $dy = [0, 1, 1, 1, 0, -1, -1, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- 이동이 끝난 뒤, 2개 이상의 파이어볼이 있는 칸에 대해 파이어볼이 합쳐진다.

		●	

파이어볼 목록			
위치(r, c)	질량(m)	속력(s)	방향(d)
(1, 3)	$5 + 7 = 12$	$2 + 1 = 3$	

차례대로 $\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow$ 에 해당
 $dx = [-1, -1, 0, 1, 1, 1, 0, -1]$
 $dy = [0, 1, 1, 1, 0, -1, -1, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- 합쳐진 파이어볼은 4개의 파이어볼로 나누어진다.

		↑ ← → ↓	

파이어볼 목록			
위치(r, c)	질량(m)	속력(s)	방향(d)
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 0
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 2
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 4
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 6

차례대로 ↑, ↗, →, ↘, ↓, ↙, ←, ↖ 에 해당
 $dx = [-1, -1, 0, 1, 1, 1, 0, -1]$
 $dy = [0, 1, 1, 1, 0, -1, -1, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

문제 해결 아이디어

코테 대비.
핵심 유형 문제풀이

- 질량이 0인 파이어볼은 소멸되어 없어진 뒤에, **전체 과정을 K 번 반복**한다.

		↑ ← → ↓	

파이어볼 목록			
위치(r, c)	질량(m)	속력(s)	방향(d)
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 0
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 2
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 4
(1, 3)	$12 / 5 = 2$	$3 / 2 = 1$	인덱스 6

차례대로 ↑, ↗, →, ↘, ↓, ↙, ←, ↖에 해당
 $dx = [-1, -1, 0, 1, 1, 1, 0, -1]$
 $dy = [0, 1, 1, 1, 0, -1, -1, -1]$

코딩 테스트 대비
핵심 유형 문제풀이

소스 코드 1)

코테 대비.

핵심 유형 문제풀이

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

# 맵의 크기(N), 파이어볼의 수(M), 이동 명령 수(K)
N, M, K = map(int, input().split())

# 초기 모든 파이어볼 정보 입력
balls = []
for i in range(M):
    x, y, m, s, d = map(int, input().split())
    balls.append((x - 1, y - 1, m, s, d))

# 차례대로 ↑, ↗, →, ↘, ↓, ↙, ←, ↖에 해당
dx = [-1, -1, 0, 1, 1, 1, 0, -1]
dy = [0, 1, 1, 1, 0, -1, -1, -1]
```

```
# 모든 파이어볼을 이동하는 함수
def move():
    # N X N 맵의 어느 위치에 몇 개의 파이어볼이 있는지 표현
    arr = [[[[] for _ in range(N)] for _ in range(N)]
    # 모든 파이어볼이 자신의 방향(d)으로 속도(s)만큼 이동
    for i in range(len(balls)):
        x, y, m, s, d = balls[i]
        x = (x + dx[d] * s) % N
        y = (y + dy[d] * s) % N
        arr[x][y].append((m, s, d)) # (질량, 속도, 방향)
        balls[i] = (x, y, m, s, d) # 파이어볼 이동
    return arr # 이동 결과 배열
```

코딩 테스트 대비 핵심 유형 문제풀이

소스 코드 2)

코테 대비.

핵심 유형 문제풀이

```
def split(): # 2개 이상의 파이어볼이 있는 칸에서 파이어볼을 분할하는 함수
    new_balls = [] # 분할되어 새롭게 추가된 파이어볼들
    removed = set() # 분할되면서 제거된 파이어볼 위치들
    for i in range(N):
        for j in range(N):
            if len(arr[i][j]) >= 2: # 2개 이상의 파이어볼이 있는 칸이라면
                removed.add((i, j))
                sum_m = 0 # 질량의 합
                sum_s = 0 # 속력의 합
                even = True # 전부 짝수인지 여부
                odd = True # 전부 홀수인지 여부
                for (m, s, d) in arr[i][j]: # 해당 위치에 있는 모든 파이어볼을 확인하며
                    sum_m += m
                    sum_s += s
                    if d % 2 == 0: odd = False
                    if d % 2 == 1: even = False
                # 해당 위치의 모든 파이어볼이 일관적으로 짝수/홀수인지 확인
                if odd or even: directions = [0, 2, 4, 6]
                else: directions = [1, 3, 5, 7]
                for d in directions: # 각 방향으로 이동하는 파이어볼 4개로 분할
                    if sum_m // 5 > 0: # 질량이 존재할 때만 추가하기
                        ball = (i, j, sum_m // 5, sum_s // len(arr[i][j]), d)
                        new_balls.append(ball)
    return removed, new_balls
```

코딩 테스트 대비
핵심 유형 문제풀이

소스 코드 3)

코테 대비.

핵심 유형 문제풀이

```
for _ in range(K):
    arr = move() # 모든 파이어볼 이동
    # 2개 이상 모인 파이어볼 쪼개기
    removed, new_balls = split()
    for ball in balls:
        x, y = ball[:2]
        # 분할되면서 제거된 파이어볼들을 제외하기
        if (x, y) not in removed:
            new_balls.append(ball)
    balls = new_balls

# 정답 출력하기
answer = 0
for ball in balls:
    answer += ball[2]
print(answer)
```