

예시)  $N = 8$  map =

0	1	3	3	3	2	1	2
0	3	1	2	1	3	0	2
0	1	2	2	0	1	1	2
0	1	2	2	0	1	1	2
0	2	0	0	3	2	2	0
1	3	3	2	3	0	2	0
2	3	1	2	0	0	1	2
0	2	3	2	2	2	2	0

출발 : (0,0)

도착 : (0,0)

이동 : 오른쪽 (0,1)

아래 (1,0)

왼쪽 (0,-1)

위 (-1,0)

→ map과 똑같은 크기의 거리배열 생성  
각 좌표에 도착하는 최단 경로 기록

## [SWEA] 2806 N-Queen

예시)  $N = 2$  board = 


 1개  
1개

row0 : 어떤 column에 놓을 지 결정 → 체스판에 표시



row1 : 어떤 column에 놓을 지 결정 → 단, 나랑 같은 column, 왼·오른 대각선에  
퀸이 없어야 놓을 수 있다

⋮

row가  $n-1$ 일때

∴ 종료조건: row == N

## [SWEA] 2115 벌꿀채취

예시)  $N = 4$   
 $M = 2$   
 $C = 13$

arr =

6	1	9	7
9	8	5	8
3	4	5	3
8	2	6	7

i) 벌통들 탐색

· 서로 같은 열에 위치

ex)  $[(0,0), (0,1)], [(0,2), (0,3)]$

· 서로 다른 열에 위치

ex)  $[(0,0), (0,1)], [(1,0), (1,1)]$

iii) 가장 이득이고 조건에 맞는 조합 생성

꿀 : 6, 7, 9, 8

벌통1 벌통2

각 벌통 채취한게 : 13

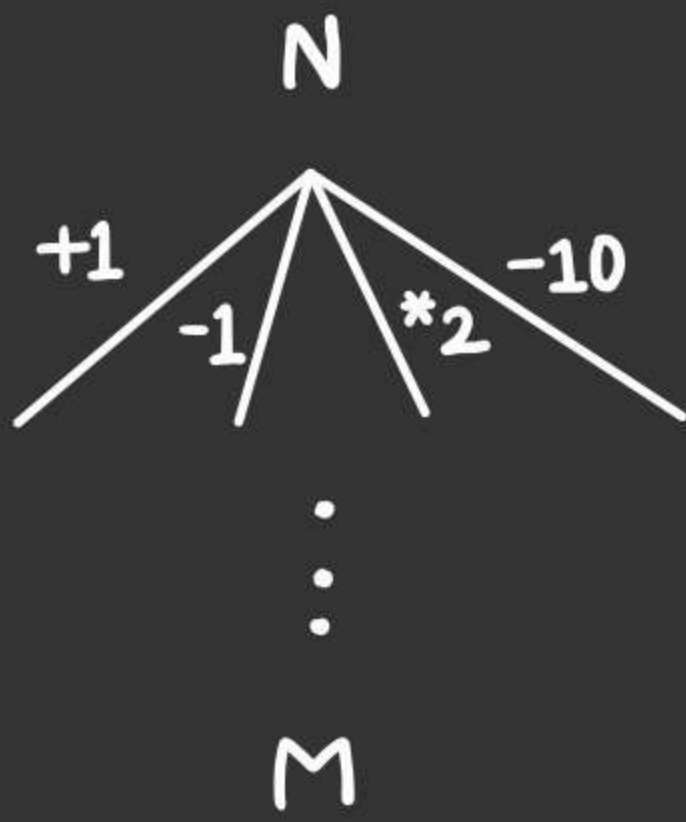
수익 : 꿀<sup>2</sup>의 합 →  $6^2 + 7^2 + 9^2$

ii) 선택한 벌통의 꿀의 값으로 변환

벌통 :  $[(0,0), (0,1), (1,0), (1,1)]$

꿀 :  $[6, 7, 9, 8]$





완탐

```
def dfs(i, total)
    i) 최소값 가지치기
    ii) if total == M:
        if i < min_i:
            min_i = i
        return
    iii) dfs(i+1, total+1)
        dfs(i+1, total-1)
        dfs(i+1, total*2)
        dfs(i+1, total-10)
```

→ 재귀error!

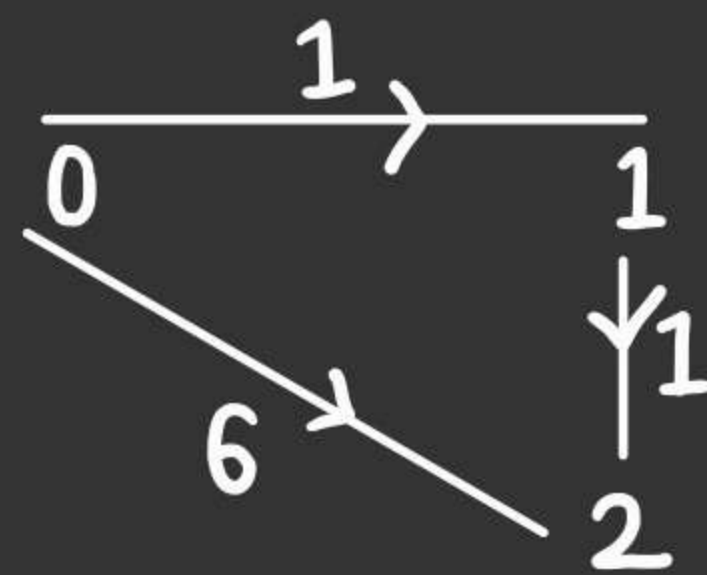
∴ bfs로 풀어보자

→ 연산결과를 정점으로 하는 방문리스트 생성  
ex)  $2+1=3$   
저장!! 4-1을 막기위함

[SWEA] 5251 최소 이동 거리

예시)  $N=2$   $M=3$  arr =

0	1	1
0	2	6
1	2	1



인접리스트  
arr (연결정점, 가중치)

	0	1
(2, 6), (1, 1)	(2, 1)	

거리리스트  
D

0	1	2
0		

0xffffffff

시작점부터 인접리스트 방문

2 → D[2]에 거리저장 ( $D[0]+6$ )  
(단, 이미 저장된 값보다 작다)

1 → D[1]에 거리저장 ( $D[0]+1$ )  
(단, 이미 저장된 값보다 작다)  
→ 인접 정점 용에 추가

(1의 인접정점)

2 → D[2]에 거리저장 ( $D[1]+1$ )  
(단, 이미 저장된 값보다 작다)

∴ 종료조건 용에 비었을 때