I 가 2개
V 가 1개
X 가 1개
L 이 0개

XIIV = 17
IIVX = 17

N^4

8/19 - 2

최백준 choi@startlink.io

로마 숫자 만들기

https://www.acmicpc.net/problem/16922

- 로마 숫자는 I, V, X, L을 사용한다. 각각의 값은 1, 5, 10, 50
- 로마 숫자를 N개 사용해서 만들 수 있는 서로 다른 수의 개수를 구하는 문제 (N ≤ 20)
- N = 1인 경우 답은 4
- N = 2인 경우 답은 10 (2, 6, 10, 11, 15, 20, 51, 55, 60, 100)

IV = 4 => 6

4^N

4가지 4 ... 4

1 2 3 4 ... N

로마 숫자 만들기

https://www.acmicpc.net/problem/16922

- 로마 숫자는 I, V, X, L을 사용한다. 각각의 값은 1, 5, 10, 50
- 로마 숫자를 N개 사용해서 만들 수 있는 서로 다른 수의 개수를 구하는 문제 ($N \le 20$)
- 경우의 수: 4^N가지

로마 숫자 만들기

https://www.acmicpc.net/problem/16922

- 로마 숫자는 I, V, X, L을 사용한다. 각각의 값은 1, 5, 10, 50
- 로마 숫자를 N개 사용해서 만들 수 있는 서로 다른 수의 개수를 구하는 문제 (N ≤ 20)
- 경우의 수: 4^N가지
- 가 아니다.
- 순서만 다른 것은 의미가 없기 때문에, 경우의 수는 N^4가지이다.

I
V
X

L : N -

N^3

로마 숫자 만들기

https://www.acmicpc.net/problem/16922

- 로마 숫자는 I, V, X, L을 사용한다. 각각의 값은 1, 5, 10, 50
- 로마 숫자를 N개 사용해서 만들 수 있는 서로 다른 수의 개수를 구하는 문제 (N ≤ 20)
- 경우의 수: 4^N가지
- 가 아니다.
- 순서만 다른 것은 의미가 없기 때문에, 경우의 수는 N^4가지이다.
- I, V, X의 개수를 알고 있다면, L의 개수도 알 수 있기 때문에, 경우의 수는 N^3가지이다.

로마 숫자 만들기

https://www.acmicpc.net/problem/16922

- 소스: http://codeplus.codes/a29092bbacdc41d5b834b5e15241068f

두 동전

https://www.acmicpc.net/problem/16197

$4^{10} =$

1 2 ... 10

4 4 ... 4

- N×M 크기의 보드, 4개의 버튼이 있다.
- 칸은 비어있거나, 동전, 벽이다.
- 동전은 2개이다.
- 버튼은 왼쪽, 오른쪽, 위, 아래이고, 누르면 그 방향으로 이동한다.
- 이동하려는 칸이 벽이면 이동하지 않는다.
- 이동하려는 칸이 없으면 보드 바깥으로 떨어진다.
- 그 외에는 이동한다.
- 두 동전 중 하나만 보드에 떨어뜨리기 위해 버튼을 몇 번 눌러야 하는가?
- 10번보다 많이 눌러야 하면 -1을 출력한다.

두 동전

https://www.acmicpc.net/problem/16197

- 총 4개의 방향을 10번까지 수행할 수 있다.
- 방법의 수: 4^{10}

두 동전

https://www.acmicpc.net/problem/16197

- go(step, x1, y1, x2, y2)
  - step: 버튼을 누른 횟수
  - (x1, y1): 한 동전의 위치
  - (x2, y2): 다른 동전의 위치

③ 다음 경우

① 정답을 찾은 경우

ㄴ> 동전 하나만 떨어진 경우

② 불가능한 경우

1 동전이 모두 떨어진 경우

2 step > 10

두 동전

https://www.acmicpc.net/problem/16197

- go(step, x1, y1, x2, y2)
    - step: 버튼을 누른 횟수
    - (x1, y1): 한 동전의 위치
    - (x2, y2): 다른 동전의 위치
- 불가능한 경우
    - step == 11
    - 동전이 둘 다 떨어진 경우
- 정답을 찾은 경우
    - 동전 하나만 떨어진 경우
- 다음 경우
    - go(step+1, nx1, ny1, nx2, ny2)

두 동전

https://www.acmicpc.net/problem/16197

- 소스: http://codeplus.codes/d3b6a84b952e405f99d33c5ebba04933

(x, y)

x행 y열

BFS

(x-1, y)

(x, y-1) ← (x, y) → (x, y+1)

(x+1, y)

dx = [-1, 1, 0, 0]

dy = [0, 0, 1, -1]

dx = [-1, 1, 0, 0, 1, 1, -1, -1]

dy = [0, 0, -1, 1, 1, -1, 1, -1]

두 스티커

$\boxed{N^2} \times 2^2$

https://www.acmicpc.net/problem/16937

- 크기가 H×W인 모눈 종이가 있고, 스티커 N개가 있다. 1 ≤ H, W, N ≤ 100
- i번 스티커의 크기는 $R_i \times C_i$이다. 스티커는 회전시킬 수 있다. 90°
- 스티커 2개를 붙이려고 한다. 두 스티커는 겹치면 안된다. 접하는 것은 가능하다.
- 붙여진 넓이의 최댓값을 구하는 문제

두 스티커

https://www.acmicpc.net/problem/16937

- 크기가 H×W인 모눈 종이가 있고, 스티커 N개가 있다. $1 \le H, W, N \le 100$
- i번 스티커의 크기는 $R_i \times C_i$이다. 스티커는 회전시킬 수 있다.
- **스티커 2개**를 붙이려고 한다. 두 스티커는 겹치면 안된다. 접하는 것은 가능하다.
- 붙여진 넓이의 최댓값을 구하는 문제
- 경우의 수 = N^2

두 스티커

https://www.acmicpc.net/problem/16937

- 크기가 H×W인 모눈 종이가 있고, 스티커 N개가 있다. $1 \leq H, W, N \leq 100$
- i번 스티커의 크기는 $R_i \times C_i$이다. 스티커는 **회전**시킬 수 있다.
- **스티커 2개**를 붙이려고 한다. 두 스티커는 겹치면 안된다. 접하는 것은 가능하다.
- 붙여진 넓이의 최댓값을 구하는 문제
- 경우의 수 = $N^2 \times 2^2$

두 스티커

$8N^2$

https://www.acmicpc.net/problem/16937

- 크기가 H×W인 모눈 종이가 있고, 스티커 N개가 있다. 1 ≤ H, W, N ≤ 100
- i번 스티커의 크기는 $R_i \times C_i$이다. 스티커는 **회전**시킬 수 있다.
- **스티커 2개**를 **붙이려고 한다**. 두 스티커는 겹치면 안된다. 접하는 것은 가능하다.
- 붙여진 넓이의 최댓값을 구하는 문제
- 경우의 수 = $N^2 \times 2^2 \times 2$

두 스티커

https://www.acmicpc.net/problem/16937

- 소스: http://codeplus.codes/4ff039beec264351bd3334060a216fc4

배열 돌리기 4

https://www.acmicpc.net/problem/17406

- 크기가 N×M인 배열 A가 있고, 배열 A의 값은 행에 있는 모든 수의 합 중 최솟값이다.
- 회전 연산은 (r, c, s) 세 정수로 이루어져 있고, 가장 왼쪽 윗 칸이 (r-s, c-s), 가장 오른쪽 아랫 칸이 (r+s, c+s)인 정사각형을 시계 방향으로 한 칸씩 돌리는 것이다.
- 회전 연산 K개의 순서를 정해서 배열 A의 값의 최솟값을 구하는 문제
- N, M ≤ 50, 1 ≤ K ≤ 6

$K!$

$6! = 720$

NM

$\underbrace{K!}_{\text{방법의 수}} \times \underbrace{NM}_{\text{회전연산 1개의 시간}} \times K$ 연산이 K개

배열 돌리기 4

https://www.acmicpc.net/problem/17406

S번째 껍질의 기준:

A[r-s][c-s

오른쪽: A[r+s][c+s]

S

- 회전 연산이 (3, 4, 2)인 경우

A[1][1]	A[1][2] →	A[1][3] →	A[1][4] →	A[1][5] →	A[1][6]
	↑				↓
A[2][1]	A[2][2]	A[2][3] →	A[2][4] →	A[2][5]	A[2][6]
	↑	↑		↓	↓
A[3][1]	A[3][2]	A[3][3]	A[3][4]	A[3][5]	A[3][6]
	↑	↑		↓	↓
A[4][1]	A[4][2]	A[4][3] ←	A[4][4] ←	A[4][5]	A[4][6]
	↑				↓
A[5][1]	A[5][2] ←	A[5][3] ←	A[5][4] ←	A[5][5] ←	A[5][6]
A[6][1]	A[6][2]	A[6][3]	A[6][4]	A[6][5]	A[6][6]

배열 돌리기 4

https://www.acmicpc.net/problem/17406

- 가능한 순서가 K! ≤ 6! = 720가지 밖에 안된다.
- 모든 순서를 다 만들고 해본다.

배열 돌리기 4

https://www.acmicpc.net/problem/17406

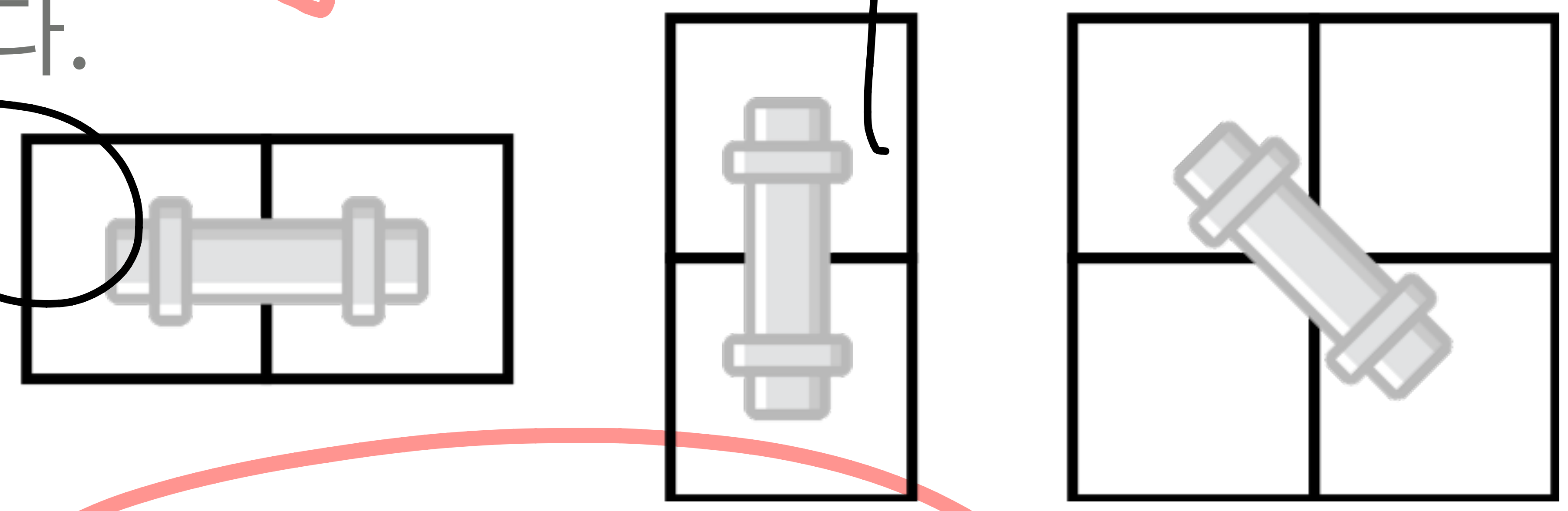
- 소스: http://codeplus.codes/e252b36bafe94c86983d74e70f096eb0

파이프 옮기기 1

A C D E F B

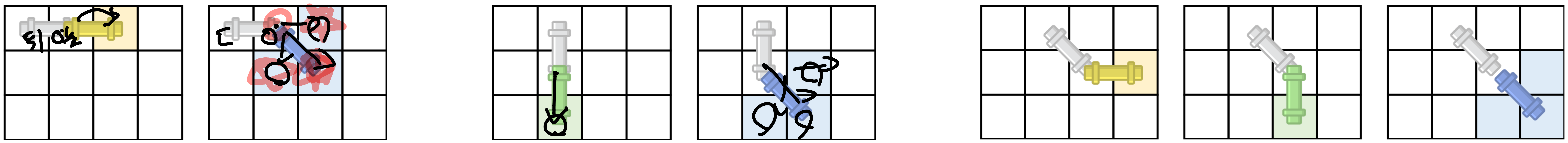
https://www.acmicpc.net/problem/17070

- N×N의 격자판으로 나타낼 수 있고, 1×1크기의 정사각형 칸으로 나누어져 있다. (N ≤ 16)
- 파이프는 2개의 연속 칸을 차지하고, 3가지 방향이 가능하다.



- 이동 가능한 방법은 총 3가지
- (1, 1), (1, 2)에 파이프가 하나 있고
- 한쪽 끝을 (N, N)으로 이동시키는 방법의 수를 구하는 문제 (방법의 수 ≤ 1,000,000)

앞 (r, c) 방향



파이프 옮기기 1

https://www.acmicpc.net/problem/17070

- 파이프가 2개의 칸을 차지하지만, 이동 가능한 방법을 보면 한쪽 끝만 저장해서 해결할 수 있다.
- 한쪽 끝의 좌표와 방향을 알고 있으면, 어디로 이동할 수 있는지 계산할 수 있다.
- 가능한 방법이 1,000,000가지 있기 때문에, 모든 방법을 다 시도해보면 된다.

파이프 옮기기 1

https://www.acmicpc.net/problem/17070

- 소스: http://codeplus.codes/776ce84b50804399885679eb69c5f60b

텔레포트

https://www.acmicpc.net/problem/16958

- 2차원 평면 위에 N개의 도시가 있다. ($2 \leq N \leq 1,000$)
- (r1, c1)에서 (r2, c2)로 가는 거리는 $|r1-r2| + |c1-c2|$이다.
- 만약, 두 도시가 특별한 도시라면, 텔레포트를 이용해서 이동할 수 있다. (시간: T)
- 두 도시의 쌍 M개가 주어졌을 때, 각각의 최소 이동 시간을 구하는 문제

텔레포트

https://www.acmicpc.net/problem/16958

- 한 도시 A에서 또다른 도시 B로 가는 방법은 다음 4가지 중 하나이다.
  - A → B
  - A → A의 근처 특별한 도시 → B (B가 특별한 도시인 경우)
  - A → B의 근처 특별한 도시 → B (A가 특별한 도시인 경우)
  - A → A의 근처 특별한 도시 → B의 근처 특별한 도시 → B

텔레포트

https://www.acmicpc.net/problem/16958

- 한 도시 A에서 또다른 도시 B로 가는 방법은 다음 2가지 중 하나이다.
  - A → B
  - A → A의 근처 특별한 도시 → B의 근처 특별한 도시 → B

텔레포트

https://www.acmicpc.net/problem/16958

- 한 도시 A에서 또다른 도시 B로 가는 방법은 다음 2가지 중 하나이다.
  - A → B
  - A → A의 근처 특별한 도시 → B의 근처 특별한 도시 → B
- O(N)에 계산할 수 있다.

텔레포트

https://www.acmicpc.net/problem/16958

- 소스: http://codeplus.codes/b4f60f189220440ab50412e71f2fa54b