

# 그리디 알고리즘 (도전)

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

<https://www.acmicpc.net/problem/1201>

- 1부터  $N$ 까지의 수를 한 번씩 이용해서 최대 부분 증가 수열의 길이가  $M$ 이고, 최대 부분 감소 수열의 길이가  $K$ 인 수열을 구하는 문제

# NMK

<https://www.acmicpc.net/problem/1201>

- 불가능한 경우 찾기
- 적어도 M개의 정수는 증가 수열에 포함되어야 하고
- 적어도 K개의 정수는 감소 수열에 포함되어야 한다
- 두 수열은 최대 정수 1개를 공유할 수 있기 때문에
- $N \geq M+K-1$  이어야 한다

# NMK

<https://www.acmicpc.net/problem/1201>

- 불가능한 경우 찾기
- 또,  $N$ 은  $MK$ 를 넘을 수 없다
- $N = MK+1$ 인 경우에 길이가  $M+1$ 인 증가 수열이나 길이가  $K+1$ 인 감소 수열을 반드시 만들 수 있다.
- 비둘기집 원리로 증명할 수 있음
- Erdős-Szekeres Theorem
- <http://mathworld.wolfram.com/Erdos-SzekeresTheorem.html>

# NMK

<https://www.acmicpc.net/problem/1201>

- $M+K-1 \leq N \leq MK$  인 경우에만 답을 찾을 수 있다

<https://www.acmicpc.net/problem/1201>

1. 1부터  $N$ 까지 수를 오름차순으로 적는다
2. 수를  $M$ 등분 한다. 이 때, 그룹에 들어있는 수는  $K$ 보다 작거나 같아야 하며, 적어도 한 그룹은 들어있는 수의 개수가  $K$ 이어야 한다
3. 각 그룹에 들어있는 수의 순서를 뒤집는다

# NMK

<https://www.acmicpc.net/problem/1201>

- $N = 13, M = 5, K = 4$  인 경우

1. 1 2 3 4 5 6 7 8 9 10 11 12 13

2. [1 2 3] [4] [5 6 7 8] [9 10] [11 12 13]

3. [3 2 1] [4] [8 7 6 5] [10 9] [13 12 11]

- 사전 순으로 가장 앞서는 순열을 찾는 경우에는
- [1] [2] [5 4 3] [9 8 7 6] [13 12 11 10]
- 와 같이 나누어야 한다

<https://www.acmicpc.net/problem/1201>

- 소스: <http://codeplus.codes/84c29b05c6cc445382658637f46a8622>



# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- $R \times C$  크기의 보드의 각 칸에 숫자가 써있다
- $(1, 1) \rightarrow (R, C)$ 로 이동하는데 방문한 칸의 숫자의 합을 최대로
- 위 아래 오른쪽 왼쪽으로 이동 가능
- 같은 칸은 한 번만 방문 가능

# 롤러코스터

10

<https://www.acmicpc.net/problem/2873>

- R 또는 C가 홀수면 모든 칸을 방문할 수 있음

# 롤러코스터

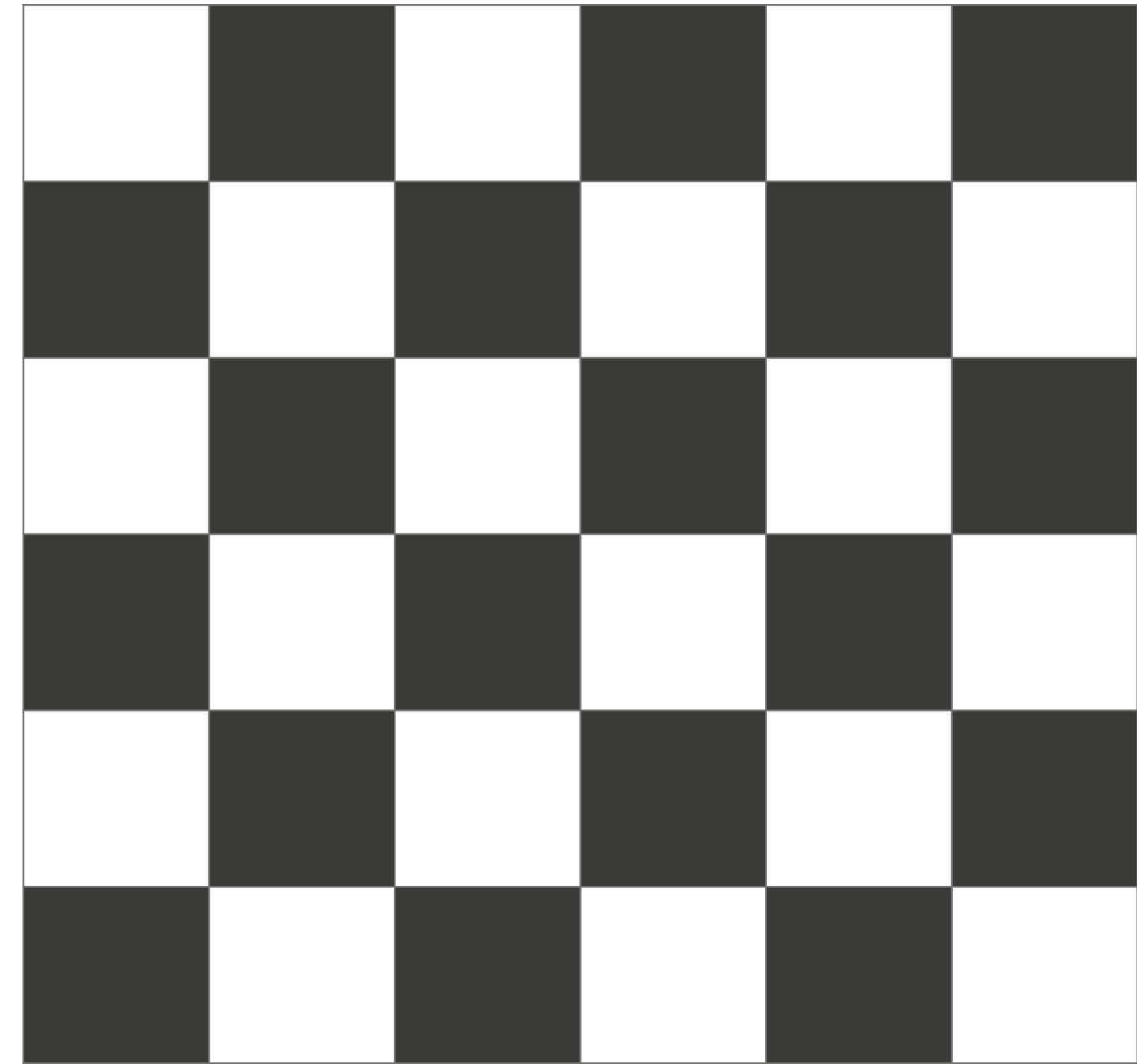
<https://www.acmicpc.net/problem/2873>

- R과 C가 모두 짝수면 모든 칸을 방문하는 것은 불가능

# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 모든 칸을 체스판 처럼 검정과 흰색으로 칠했다고 하자
- (1, 1)과 (R, C)의 색은 흰색이다
- (1, 1)과 (R, C)로 가는 모든 경로는  
흰 → 검 → 흰 → ... → 흰 → 검 → 흰
- 방문한 칸은 흰>검이다.
- 방문하지 않은 칸 흰<검
- 따라서, 모든 칸을 방문하는 것이 불가능



# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 흰 칸 한 칸을 방문하지 않는다면, 나머지 칸은 모두 방문 불가
- 검정 칸 한 칸을 방문하지 않으면, 나머지 칸을 모두 방문 가능
- 따라서, 방문하지 않을 검정 칸 하나를 선택해야 함
- 방문한 칸의 합의 최대를 구하는 문제이기 때문에, 가장 작은 값을 가지는 검정 칸을 선택!

# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 문제를 변형해서 풀기
- 두 사람이  $(1, 1)$ ,  $(R, C)$ 에 있고, 서로 만날때까지 이동하는 문제
- $(1, 1)$ 에 있는 사람을 A,  $(R, C)$ 에 있는 사람을 B

# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 선택한 칸이 첫 두 행에 없다면
  - A는 첫 행의 오른쪽으로 갔다가 아래로 한 칸 내려오고 두 번째 칸의 왼쪽으로 이동한 다음, 한 칸 아래로 내려온다
  - 이렇게 되면, 위의 두 행을 무시하고 다시 문제를 풀 수 있다.
- 
- 선택한 칸이 마지막 두 행에 없다면
  - 위와 같은 식으로 B를 이동시켜
  - 마지막 두 행을 무시하고 문제를 다시 풀 수 있다

# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 이런식으로 하면, 행은 2개만 남게 되고
- 여기서부터는 열을 행과 같은 식으로 처리하면
- 결국  $2 \times 2$  크기의 칸만 남게 된다



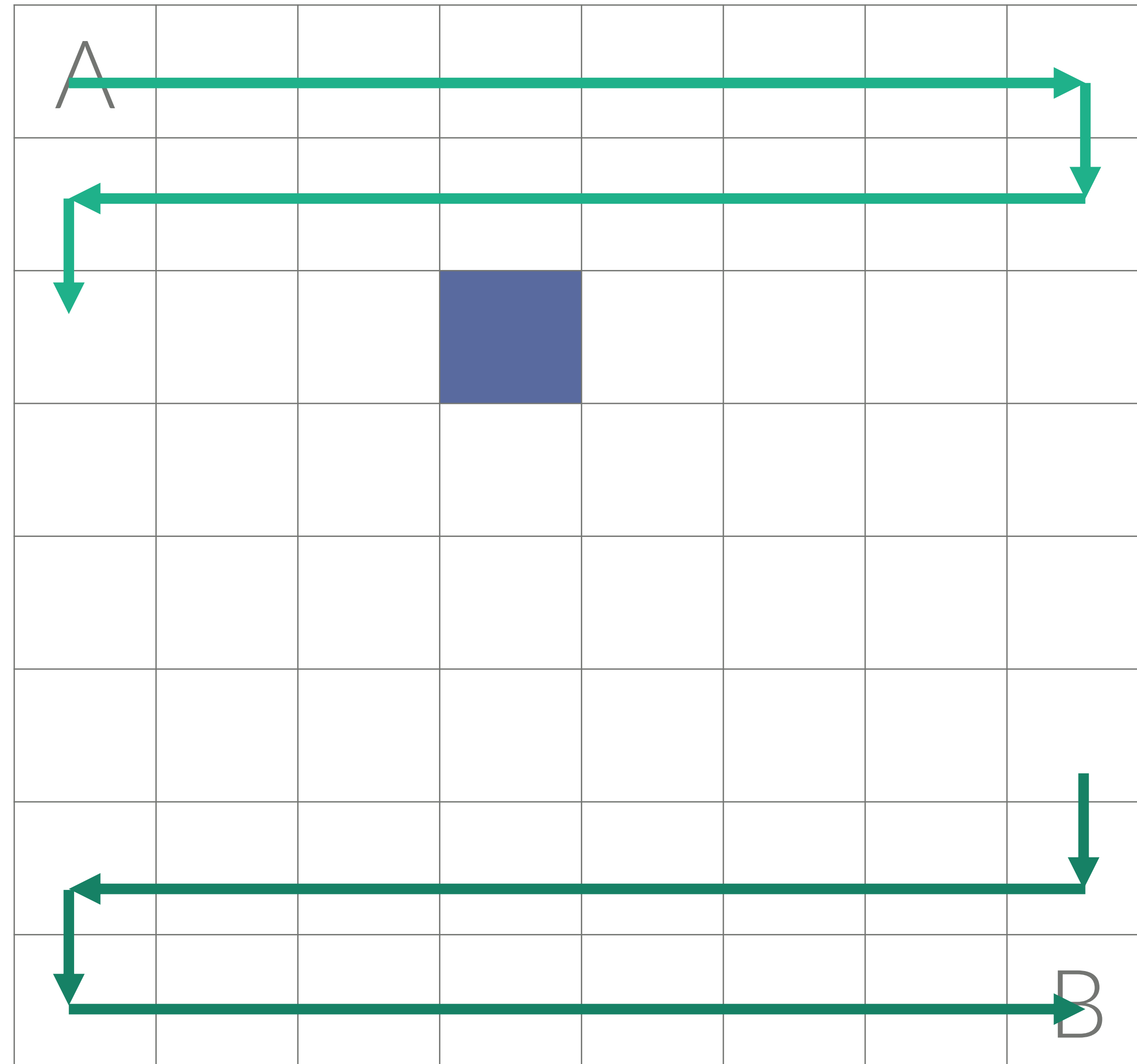
# 롤러코스터

<https://www.acmicpc.net/problem/2873>

[illegible]

# 롤러코스터

<https://www.acmicpc.net/problem/2873>



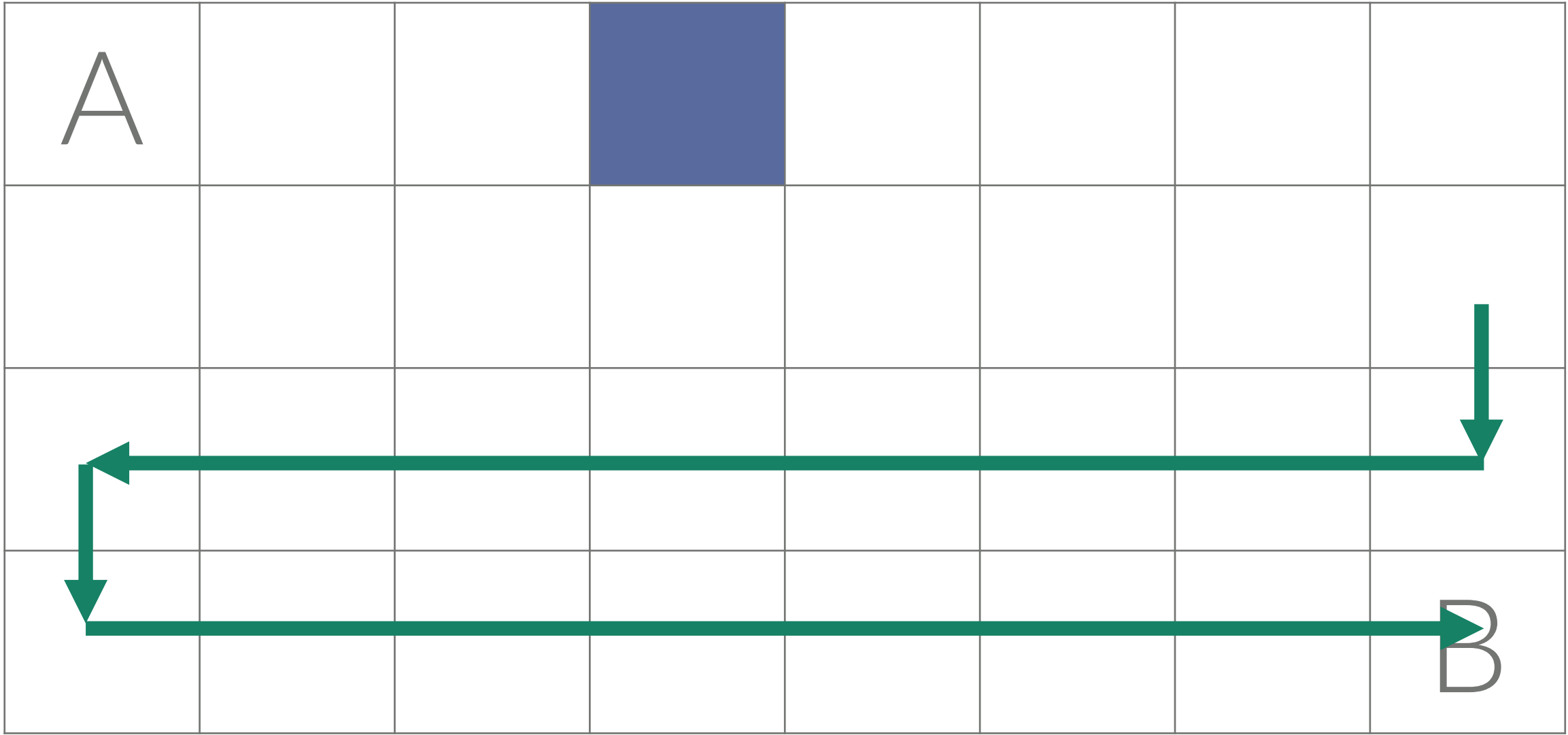
# 롤러코스터

<https://www.acmicpc.net/problem/2873>

A							
							B

# 롤러코스터

<https://www.acmicpc.net/problem/2873>



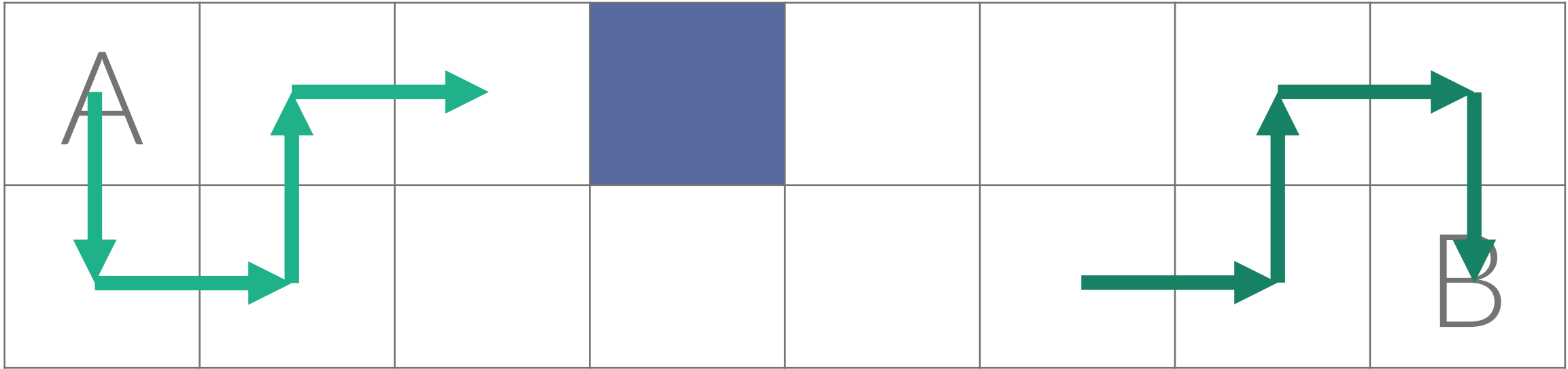
# 롤러코스터

<https://www.acmicpc.net/problem/2873>

A							
							B

# 롤러코스터

<https://www.acmicpc.net/problem/2873>



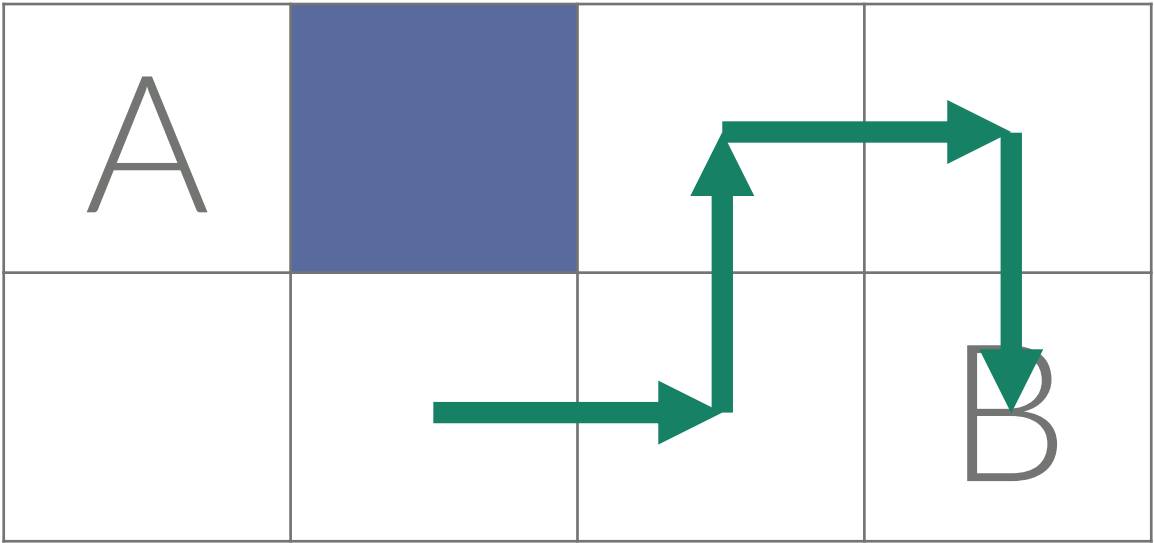
# 롤러코스터

<https://www.acmicpc.net/problem/2873>

A			
			B

# 롤러코스터

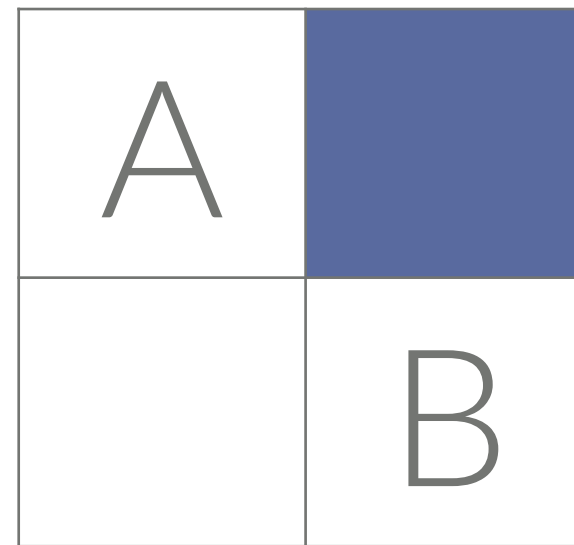
<https://www.acmicpc.net/problem/2873>





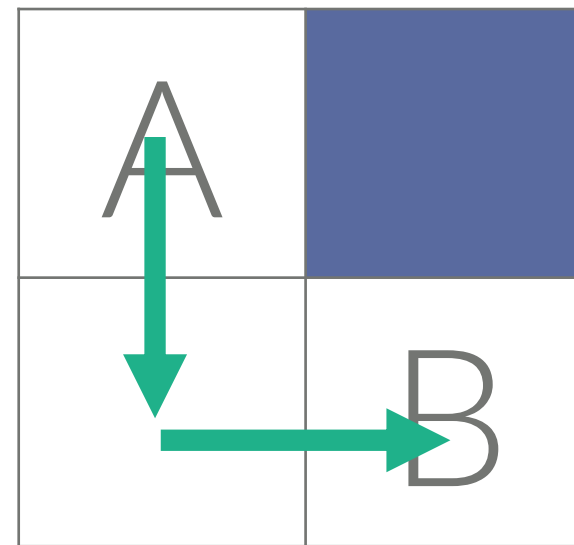
# 롤러코스터

<https://www.acmicpc.net/problem/2873>



# 롤러코스터

<https://www.acmicpc.net/problem/2873>



# 롤러코스터

<https://www.acmicpc.net/problem/2873>

- 소스: <http://codeplus.codes/cbf062261c29485a9fb6e5a47ce0a115>

# A와 B 2

<https://www.acmicpc.net/problem/12919>

- S를 T로 바꾸는 문제
- 가능한 연산
- 문자열의 뒤에 A를 추가한다
- 문자열을 뒤에 B를 추가하고 문자열을 뒤집는다
- $1 \leq S$ 의 길이  $\leq 49$ ,  $2 \leq T$ 의 길이  $\leq 50$ ,  $S$ 의 길이  $< T$ 의 길이
- $S = "A"$ ,  $T = "BABA"$ , 정답: 1
- $S = "A"$ ,  $T = "ABBA"$ , 정답: 0

# A와 B 2

<https://www.acmicpc.net/problem/12919>

- T의 마지막 문자가 A라면, A연산을 사용해서 T를 만든 것이다.
- T의 첫 문자가 B라면, B연산을 사용해서 T를 만든 것이다.

# A와 B 2

<https://www.acmicpc.net/problem/12919>

- T의 첫 문자가 B이고 마지막 문자가 A이면, 두 경우 모두 조사하면 된다.

# A와 B 2

<https://www.acmicpc.net/problem/12919>

- T의 첫 문자가 B이고 마지막 문자가 A이면, 두 경우 모두 조사하면 된다.
- BX.....YA 인 경우
- A연산을 되돌리면 BX.....Y가 된다.
- Y가 A인 경우 다시 두 경우 모두 조사해야 하고, B인 경우 B연산을 되돌리는 경우이다.
- B연산을 되돌리면 AY.....X가 된다.
- X가 A이면 A연산을 되돌리는 경우이고, B인 경우는 불가능한 경우이다.

# A와 B 2

<https://www.acmicpc.net/problem/12919>

- 두 방법으로 나누어지는 경우가 총  $N-1$ 번 있다.
- 모든 단계에서 문자열의 길이가 1씩 감소하기 때문에, 총 가능한  $(S, T)$ 의 조합은  $N^2$ 개가 있다.
- 문자열의 연산은  $O(N)$ 이기 때문에, 시간 복잡도는  $O(N^3)$ 이다.



# A와 B 2

<https://www.acmicpc.net/problem/12919>

- 소스: <http://codeplus.codes/5413177201c1464d8eb46df3ffecf61e>