

2023 PNU CodeRace

문제 해설

PULSE
부산대학교 알고리즘 동아리


se;lon



부산대학교
정보컴퓨터공학부


STARTLINK

 **한빛미디어**
Hanbit Media, Inc.

구분	문제	의도한 난이도	출제자
Beginner	침탑 밀어서 부수기	Easy	tykr0001
	영역 색칠	Easy	ido_013
	경품 추천	Medium	ido_013
공통	게임을 클리어하자	Medium	mbae059
Advanced	시간이 겹칠까?	Medium	mbae059
	산지니의 여행계획	Hard	mbae059
	Attention	Hard	kmiiiaa

검수자 : jthis, leeh18, shjohw12

침탑 밀어서 부수기

알고리즘 구분: 구현

의도한 난이도: **Easy**

- 출제자: tykr0001

첨탑 밀어서 부수기

- 첨탑의 개수 N 에 대해서 몇 번 밀어야 모두 넘어뜨릴 수 있는지 묻는 문제입니다.
- 첨탑의 높이를 입력받을 때마다, 바로 이전 첨탑의 높이보다 작은지 검사하면 됩니다.
- 맨 처음이거나, 바로 이전 첨탑의 높이가 크거나 같은 경우를 세어주면 됩니다.

영역 색칠

알고리즘 구분: 구현

의도한 난이도: **Easy**

- 출제자: ido_013

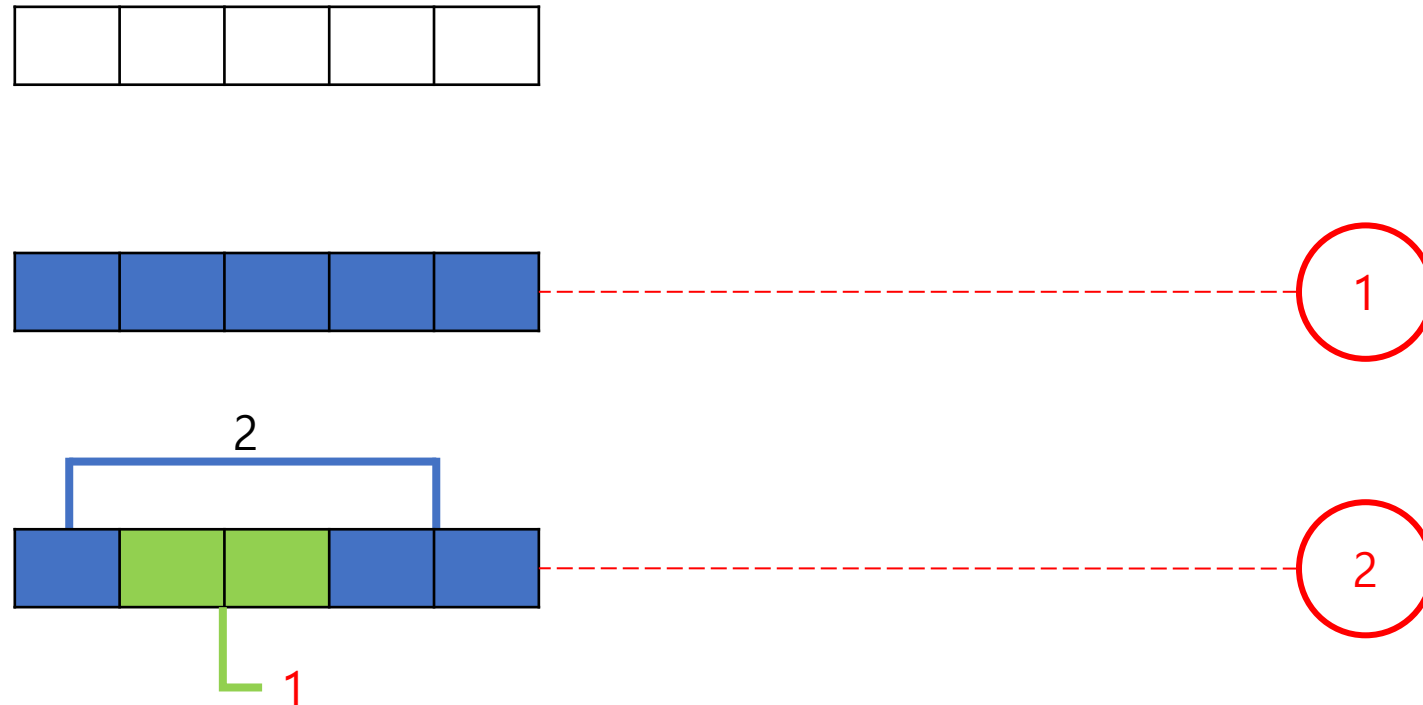
영역 색칠

- 주어진 전체 배열을 순회하면서 각 행마다 필요한 붓질 횟수를 구해 이들의 합을 구하는 문제입니다.
- 붓의 두께가 1칸, 가로 방향으로만 칠할 수 있으므로 같은 열의 색들은 붓질 횟수에 영향을 주지 않습니다.
- 어떤 행에서 같은 색으로 이어진 것을 그 색의 구간이라 하면, 구간의 길이는 붓질 횟수에 영향을 주지 않습니다.

영역 색칠

- 색이 연속적으로 칠해져 있는 영역의 최소 붓질 횟수는 ' $\min(\text{파란색 구간의 수}, \text{초록색 구간의 수}) + 1$ ' 입니다.

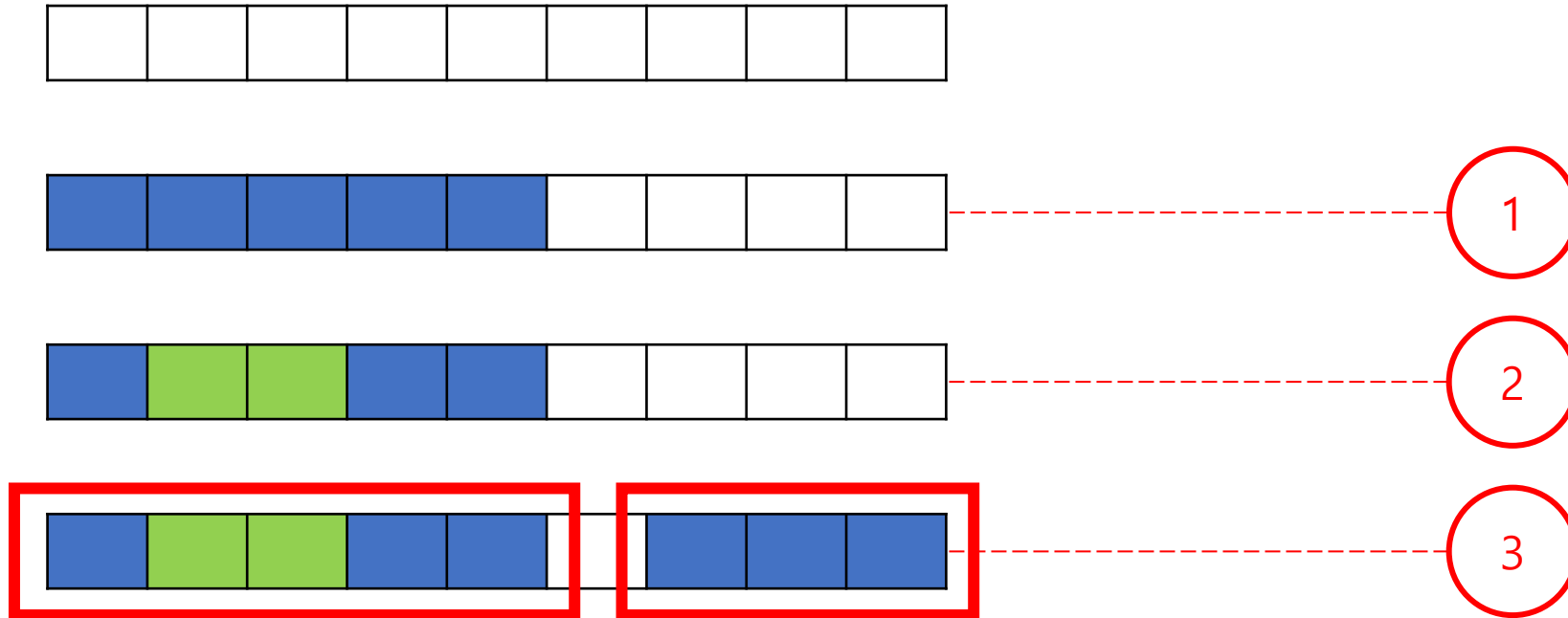
예시 1



영역 색칠

- 색이 칠해지지 않는 칸을 기준으로 왼쪽 묶음, 오른쪽 묶음이라 하면 '왼쪽 묶음 붓질 횟수 + 오른쪽 묶음 붓질 횟수' 와 같습니다. 색이 없는 칸들을 기준으로 나누어 독립적으로 붓질 횟수를 구하면 됩니다.

예시 2



경품 추첨

알고리즘 구분: Breadth-First Search

의도한 난이도: **Medium**

- 출제자: ido_013

경품 추첨

- 추첨판에서 공의 경로를 BFS로 생성하며, 각 경로에서의 확률 값을 저장하는 문제입니다.
- 못을 만나지 않으면 1, 한 번 만나면 $\frac{1}{2}$, 두 번 만나면 $\frac{1}{4}$ 와 같이 확률은 절반이 됩니다.
- 실수 자료형이 확률을 충분히 담을 수 없기에, 확률을 p[100] 형태의 배열로 저장합니다.

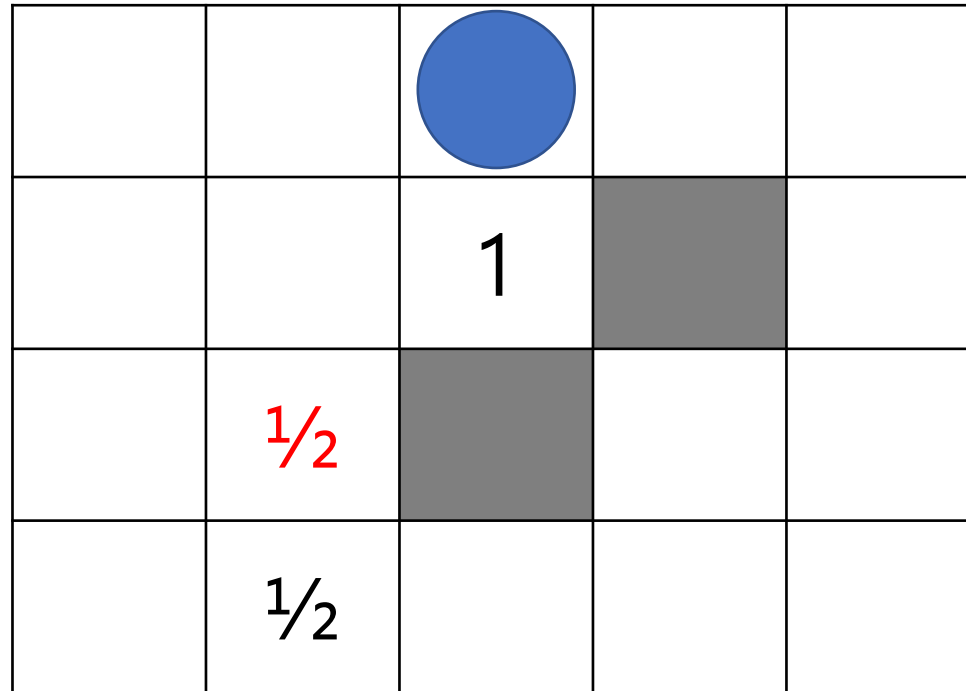
• 1	1	0	0	0	0	0	...	0	0
• $\frac{1}{2}$	0	1	0	0	0	0	...	0	0
• $\frac{3}{4}$	0	1	1	0	0	0	...	0	0

0
100

경품 추첨

- 공이 걸리는 확률 또한 절반입니다. 따라서 걸린 곳의 반대편 경로에 절반이 된 확률이 저장됩니다.

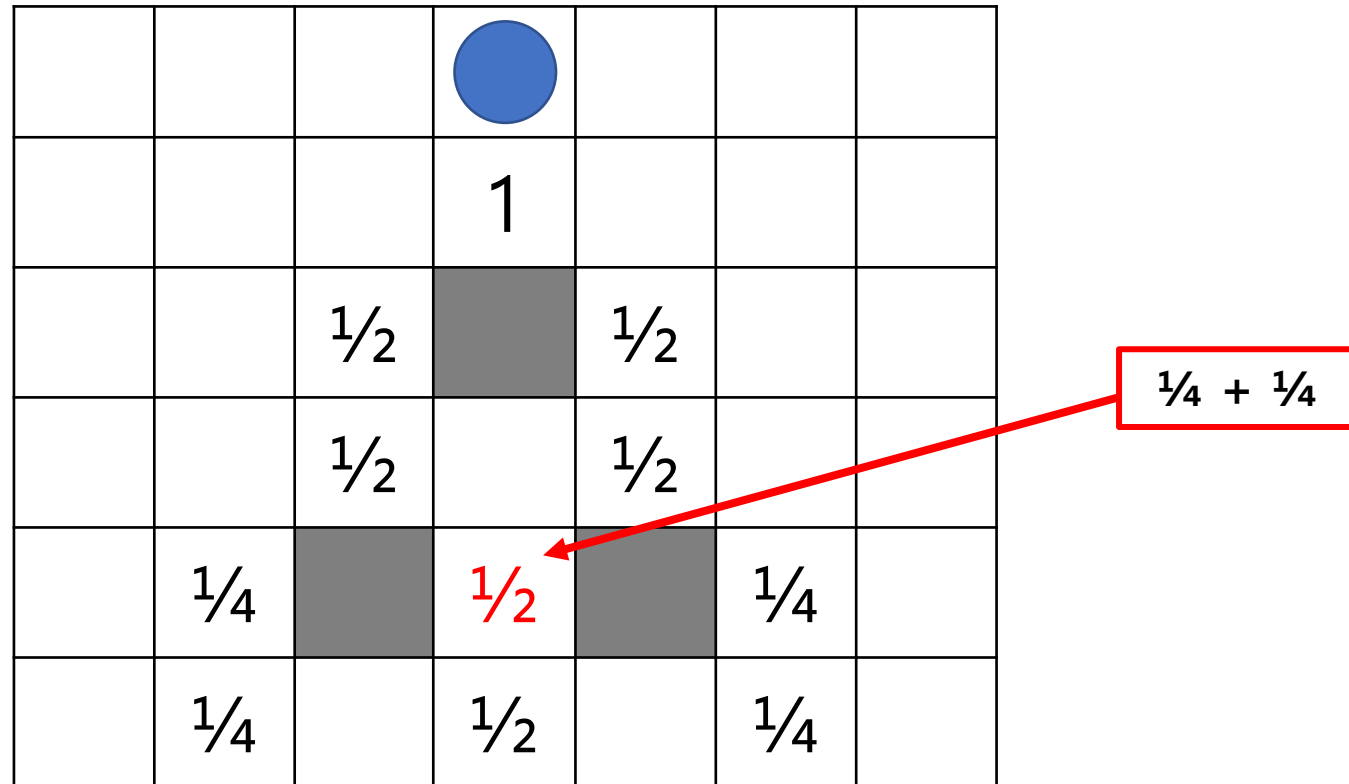
예시 1



경품 추첨

- 공의 이동 경로가 겹치는 경우에는 확률 값이 더해집니다.
- 가능한 모든 경로를 확인한 후에 가장 확률이 높은 번호를 출력합니다.

예시 2



게임을 클리어하자

알고리즘 구분: Dynamic Programming

의도한 난이도: **Medium**

- 출제자: mbae059

게임을 클리어하자

- 바로 이전 회차에 사용했던 무기를 제외한 모든 무기를 사용할 수 있습니다.
- 빠르게 클리어하기 위해서는 각 무기의 효율을 탐색할 필요가 있습니다.
- DP 테이블을 2차원 배열로 선언하고, 또 다른 2차원 배열인 Matrix에는 입력 받은 값을 저장합니다.
- 아래와 같은 점화식을 이용하여 DP 테이블을 채웁니다.
- for $k \leftarrow 1$ to M

$$DP[N][M] = \min(DP[N][M], DP[N-1][k] + \text{Matrix}[N][M]), \text{ (단, } k \neq M \text{)}$$

게임을 클리어하자

- for $k \leftarrow 1$ to M
 $DP[N][M] = \min(DP[N][M], DP[N-1][k] + Matrix[N][M]),$ (단, $k \neq M$)

예시

Matrix

1	10	10	10	10
10	10	1	10	10
10	1	10	10	10
10	10	10	1	10
10	10	10	10	1



DP table

1	10	10	10	10
20	11	2	11	11
12	3	21	12	12
13	22	13	4	13
14	14	14	23	5

시간이 겹칠까?

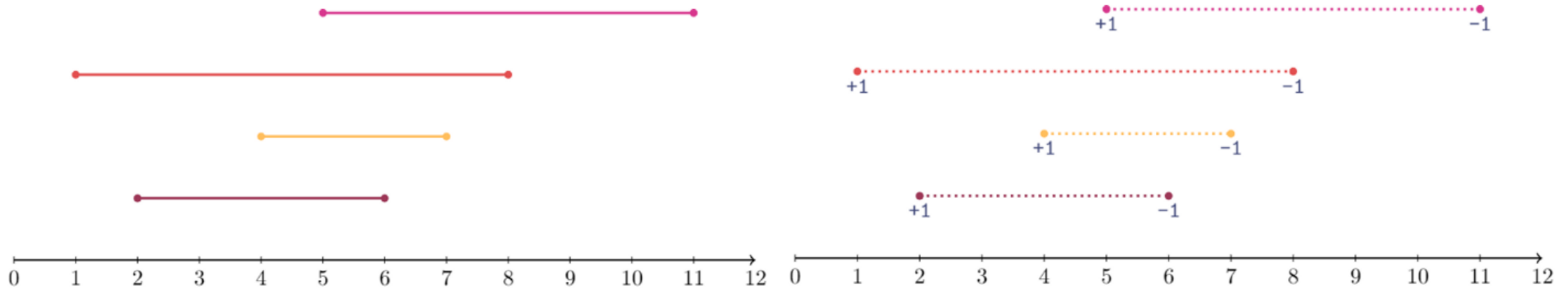
알고리즘 구분: imos

의도한 난이도: **Medium**

- 출제자: mbae059

시간이 겹칠까?

- imos법 (いもす法, imos法)
- 입력에 대하여 시작과 끝만 기록한 후, 누적합으로 1~1,000,000 까지 순회합니다.

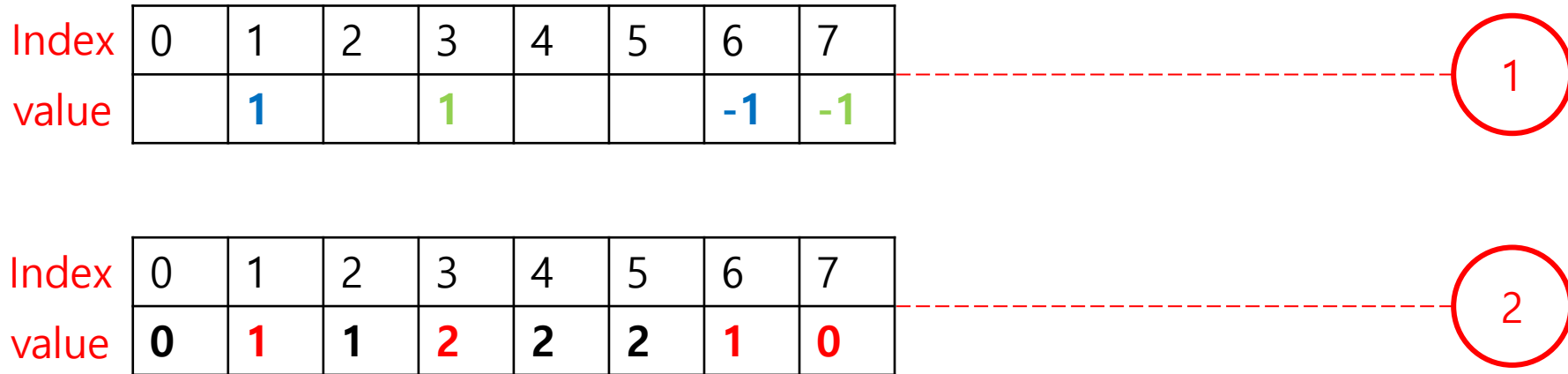


시간이 겹칠까?

- 주어진 입력에 따라 시작 시각에는 +1, (마지막 시각 + 1)에는 -1을 더하여 처리한 후, 최대 시각까지 누적합으로 진행하면 됩니다.

각 좌석은 사용이 종료되는 시각에 곧바로 선택될 수 없다.

예시 [1,5], [3,6] 이 입력으로 주어진 경우



산지니의 여행계획

알고리즘 구분: Maximum Spanning Tree, DFS

의도한 난이도: **Hard**

- 출제자: mbae059

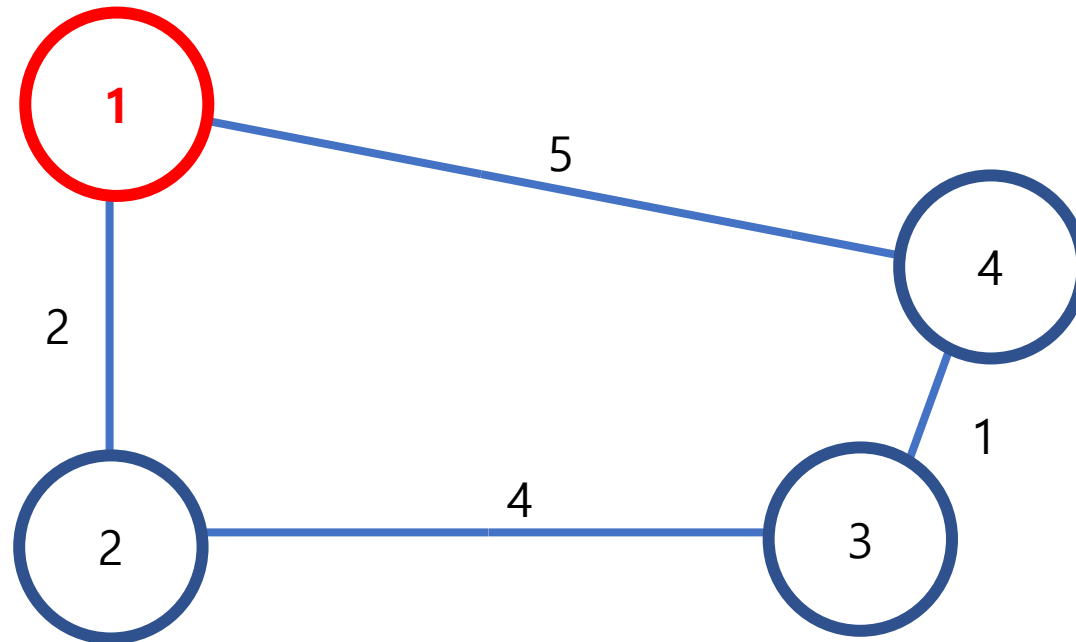
산지니의 여행계획

- **최대한 도로를 적게 선택하며** N개의 모든 도시를 방문합니다.
- **위 조건을 만족하면서** 도로 길이의 합이 최대가 되게 합니다.
→ 최대 스패닝 트리
- **선택된 도로를 토대로** 최적의 경로를 구하여 운전 거리를 줄입니다.
→ DFS
- 마지막 도시에서 시작도시로 되돌아가지 않습니다.
- **(트리를 만들 때 필요한 도로들의 합) * 2 - DFS로 구한 루트에서 리프 노드 간 최장 거리**

산지니의 여행계획

예시

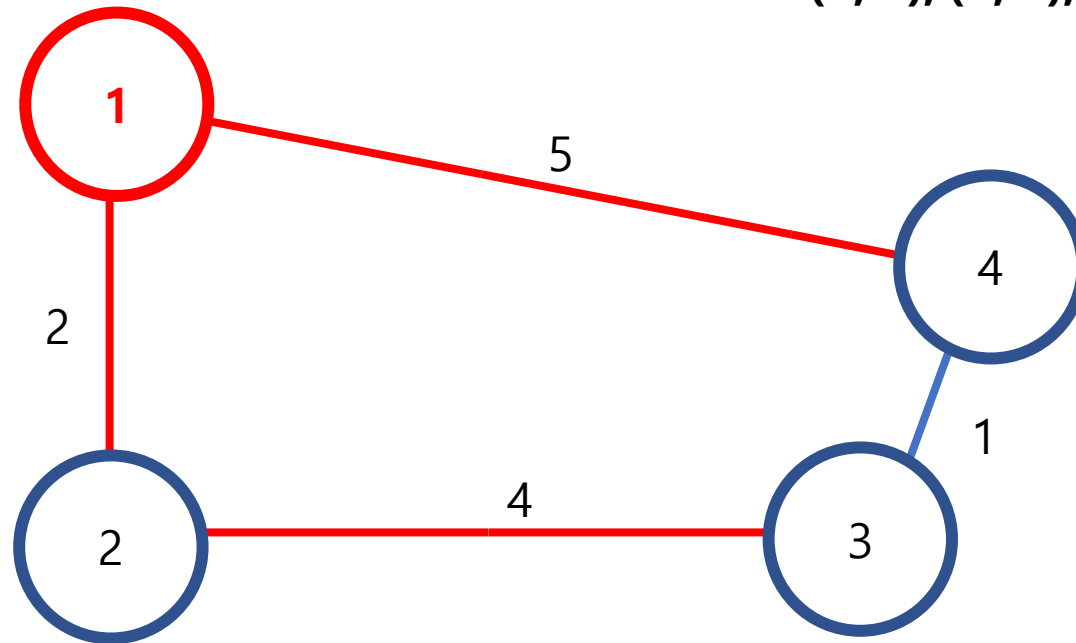
- 시작 도시 번호: 1



산지니의 여행계획

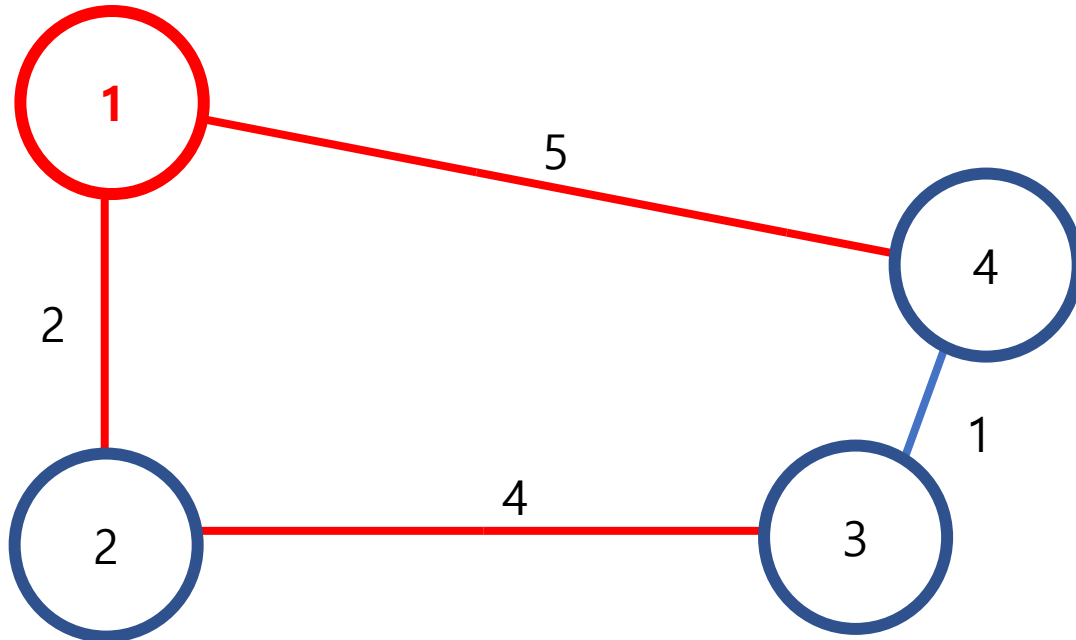
예시

- 시작 도시 번호: 1
- (1, 2), (1, 4), (2, 3) 직통 도로 선택



산지니의 여행계획

예시



- 시작 도시 번호: 1
- (1, 2), (1, 4), (2, 3) 직통 도로 선택
- (선택한 도로의 길이의 합) * 2
- = (5 + 2 + 4) * 2
- DFS 진행 - 루트에서 리프노드 간 거리 비교
- (2 + 4) > 5
- **Answer = (5 + 2 + 4) * 2 - (2 + 4) = 16**

Attention

알고리즘 구분: Fenwick Tree(Binary Indexed Tree, BIT)

의도한 난이도: **Hard**

- 출제자: kmiiiiaa

Attention

- 레시피 북 A, B 모두 참조하는 것이 아닌 1개의 배열만 참조할 수 있도록 변경합니다.
- STEP:
 1. 레시피 북 A의 원소들을 정렬할 때 본래의 인덱스가 저장되는 인덱스 배열 C 생성
 2. 레시피 북 B의 각 원소들의 인덱스 배열 C에서의 위치를 순서대로 저장하는 배열 D 생성
- 배열 D를 앞에서부터 순회하여 가능한 쿠키 레시피를 찾아냅니다.
- 가능한 쿠키 레시피
 - 이전의 원소들 중에 현재 원소보다 작으면서 증가하는 원소쌍의 개수

Attention

- 2개의 Fenwick Tree를 활용하여 배열 D 를 순회하며 원소쌍의 개수를 찾아냅니다.
- 첫번째 Fenwick Tree는 배열 D 의 현재 원소의 존재 여부를 표시합니다.
- 두번째 Fenwick Tree는 첫번째 Fenwick Tree를 참조합니다.
- 이전에 나타난 원소들 중, 현재 원소보다 작은 원소들의 개수를 저장합니다.

감사합니다.