

# 다이나믹 프로그래밍 3 (도전)

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

---

# 등차수열

<https://www.acmicpc.net/problem/1994>

- $N(1 \leq N \leq 2,000)$ 개의 음 아닌 정수들이 있다
- 이들 중 몇 개의 정수를 선택하여 나열하면 등차수열을 만들 수 있다
- 예를 들어 4, 3, 1, 5, 7이 있을 때 1, 3, 5, 7을 선택하여 나열하면 등차수열이 된다
- 이와 같이 했을 때, 등차수열의 길이가 최대 얼마까지 가능한지 알아내시오.

# 등차수열

<https://www.acmicpc.net/problem/1994>

- $D[i][j] = A[j] - A[i]$ 를 공차로 하면서 만들 수 있는 가장 큰 등차수열의 길이

# 등차수열

<https://www.acmicpc.net/problem/1994>

- $D[i][j] = D[j][next] + 1$  (next가 있는 경우,  $a[next] - a[j] == a[j] - a[i]$ )
- next는 이분 탐색을 이용해서 구할 수 있다.

# 등차수열

<https://www.acmicpc.net/problem/1994>

- 소스: <http://codeplus.codes/3c0e616f319440e9afd182cb92790f9f>

# 선물 전달

<https://www.acmicpc.net/problem/1947>

- N명의 사람이 서로 선물을 교환한다. 이 때, 자기 자신의 선물을 받는 경우는 없다.

# 선물 전달

<https://www.acmicpc.net/problem/1947>

- N명의 사람이 서로 선물을 교환한다. 이 때, 자기 자신의 선물을 받는 경우는 없다.
- N번째 사람이 i번 사람의 선물을 받고, i번 사람이 N번 사람의 선물을 받은 경우
  - 남은 사람의 수는 N-2명이기 때문에
  - $D[N-2]$
- i번 사람이 N번 사람의 선물을 받았는데, N번 사람이 다른 사람의 선물을 받은 경우
  - 남은 사람의 수는 N-1명이기 때문에,
  - $D[N-1]$

# 선물 전달

<https://www.acmicpc.net/problem/1947>

- N명의 사람이 서로 선물을 교환한다. 이 때, 자기 자신의 선물을 받는 경우는 없다.
- N번째 사람이 i번 사람의 선물을 받고, i번 사람이 N번 사람의 선물을 받은 경우
  - 남은 사람의 수는 N-2명이기 때문에
  - $D[N-2]$
- i번 사람이 N번 사람의 선물을 받았는데, N번 사람이 다른 사람의 선물을 받은 경우
  - 남은 사람의 수는 N-1명이기 때문에,
  - $D[N-1]$
- 가능한 i번 사람의 수는 N-1명이기 때문에  $D[N] = (N-1) \times (D[N-1] + D[N-2])$



# 선물 전달

<https://www.acmicpc.net/problem/1947>

- 이 수열은 완전 순열 (Complete Permutation) 또는 교란 (Dearrangement) 이라고도 한다.

# 선물 전달

10

<https://www.acmicpc.net/problem/1947>

- 소스: <http://codeplus.codes/b141c1853f2843dbbfac56ce8a7d8bcd>

# 집합의 개수

<https://www.acmicpc.net/problem/2092>

- 1부터  $T(1 \leq T \leq 200)$ 까지의 범위에 있는 수들이 총  $A$ 개 있다
- 이들 중  $K$ 개를 골라서 집합을 만들 때, 가능한 집합의 개수를 세려 한다
- 단,  $K$ 의 범위는  $1 \leq S \leq K \leq B \leq A$ 로 한다
- 즉, 두 정수  $S, B$ 를 입력받아서  $K=S$ 일 경우,  $\dots$ ,  $K=B$ 일 경우의 집합의 개수를 모두 더하려고 한다.

# 집합의 개수

<https://www.acmicpc.net/problem/2092>

- $T = 3$ , 수 = 1, 1, 2, 2, 3인 경우
- $K=1$  : {1}, {2}, {3}
- $K=2$  : {1, 1}, {1, 2}, {1, 3}, {2, 2}, {2, 3}
- $K=3$  : {1, 1, 2} {1, 1, 3} {1, 2, 2} {1, 2, 3} {2, 2, 3}
- $K=4$  : {1, 2, 2, 3} {1, 1, 2, 2} {1, 1, 2, 3}
- $K=5$  : {1, 1, 2, 2, 3}
- $S = 2$ ,  $B = 3$ 이면 정답은 10

# 집합의 개수

<https://www.acmicpc.net/problem/2092>

- $D[T][K]$  = 1부터 T까지의 수를 이용해서 크기가 K인 집합을 만드는 경우의 수
- $D[T][K] += D[T-1][K-L]$  (L은 T를 몇 번 이용할 건지를 나타냄)

# 집합의 개수

<https://www.acmicpc.net/problem/2092>

- 소스: <http://codeplus.codes/bfaecc2cf89548139853e72b36a2fa5f>

# 팰린드롬 경로

<https://www.acmicpc.net/problem/2172>

- $N \times N$  행렬에 0부터 9까지 숫자가 적혀있다.  $N \leq 20$
- 임의의 한 칸에서 시작해서 매번 인접(상하좌우, 대각선)한 칸 중 하나로 이동해서
- 총  $L$ 개의 칸을 이동한다.  $L \leq 20$
- 경로에 적힌 수를 나열한 것이 팰린드롬이 되는 경로의 개수를 구하는 문제

# 팰린드롬 경로

<https://www.acmicpc.net/problem/2172>

- $D[N][x1][y1][x2][y2]$  = (x1,y1)에서 시작해서 (x2,y2)에서 도착하는 길이가 N인 팰린드롬 경로의 개수
- 길이가 1인 팰린드롬 경로
  - $D[N][x][y][x][y] = 1$
- 자기 자신은 1이기 때문
- 길이가 2인 팰린드롬 경로
  - $D[N][x1][y1][x2][y2] = (A[x1][y1] == A[x2][y2])$
  - 이때, (x1,y1)에서 (x2,y2)로 이동할 수 있어야 함



# 팰린드롬 경로

<https://www.acmicpc.net/problem/2172>

- $D[N][x1][y1][x2][y2] += D[N-2][x3][y3][x4][y4]$ 
  - 필요한 조건
  - $A[x1][y1] == A[x2][y2]$
  - $(x1, y1), (x3, y3)$ 은 인접해야 함
  - $(x2, y2), (x4, y4)$ 도 인접해야 함

# 팰린드롬 경로

<https://www.acmicpc.net/problem/2172>

- 소스: <http://codeplus.codes/c5ca75218c6745c68930df41c0c2ed6c>

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- 정점 N개로 이루어진 그래프가 주어지고, 간선에는 알파벳 소문자가 하나씩 쓰여 있다.
- 0번에서 1번으로 가는 그래프의 보행 중에서 길이가 가장 짧은 것을 찾는 문제
- 보행: 같은 정점, 간선을 여러 번 방문할 수 있는 경로
- $N \leq 20$

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- 간선으로 만들 수 있는 팰린드롬은  $ab.....ba$  와 같은 형식이다.
- 꼭 첫 번째 글자와 마지막 글자가 같아야 한다.
- 첫 글자와 마지막 글자가 없어도 팰린드롬이 되어야 한다.

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- 처음으로 골라야 하는 두 정점  $a, b$ 는  $0 \rightarrow a, b \rightarrow 1$  간선이 있고, 두 간선에 쓰여 있는 글자가 같아야 한다

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- $D[i][j]$  =  $i$ 번 정점에서 시작해서  $j$ 번 정점으로 가는 팰린드롬 경로의 최소 길이
- $i = j$  이면 0
- $i$ 와  $j$  사이에 간선이 있으면 1

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- $D[i][j]$  =  $i$ 번 정점에서 시작해서  $j$ 번 정점으로 가는 팰린드롬 경로의 최소 길이
- 이제 두 정점  $a, b$ 를 찾는다
- $i \rightarrow a \rightarrow \dots \rightarrow \dots \rightarrow b \rightarrow j$
- 이렇게 되면  $D[a][b] + 2$ 가 최소 길이가 된다

# 팰린드롬 보행

<https://www.acmicpc.net/problem/12950>

- 소스: <http://codeplus.codes/2539bfcd377c4f60b1a038052aaeeda6>

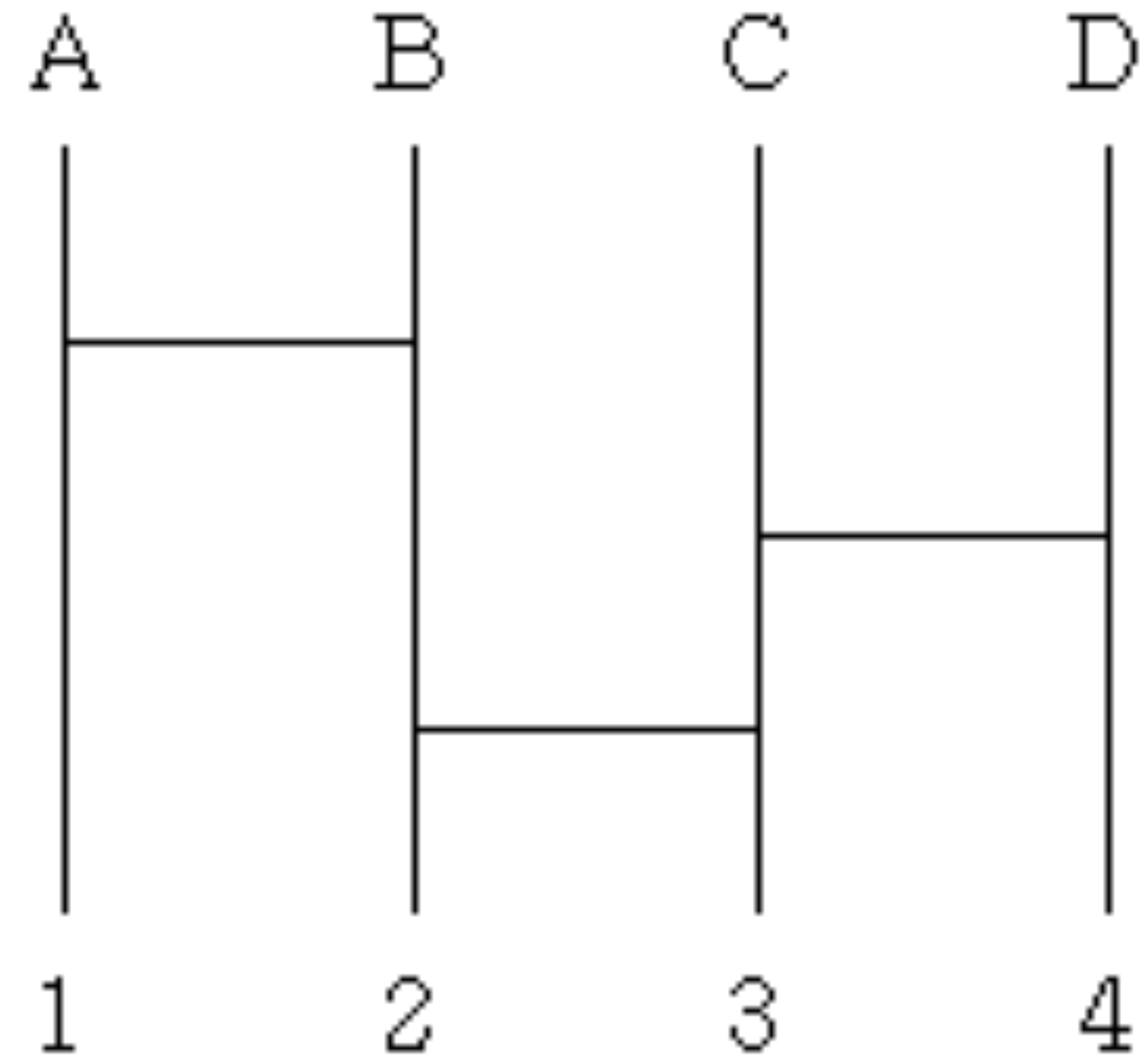


# 사다리 게임

25

<https://www.acmicpc.net/problem/2008>

- 세로선은 N개, 가로선은 M개
- a번째 위치에서 b번째 위치로 갈 수 있게 하기 위해
- 조작하는 최소 비용
- 가로선을 지우는건 비용 X
- 그리는건 비용 Y

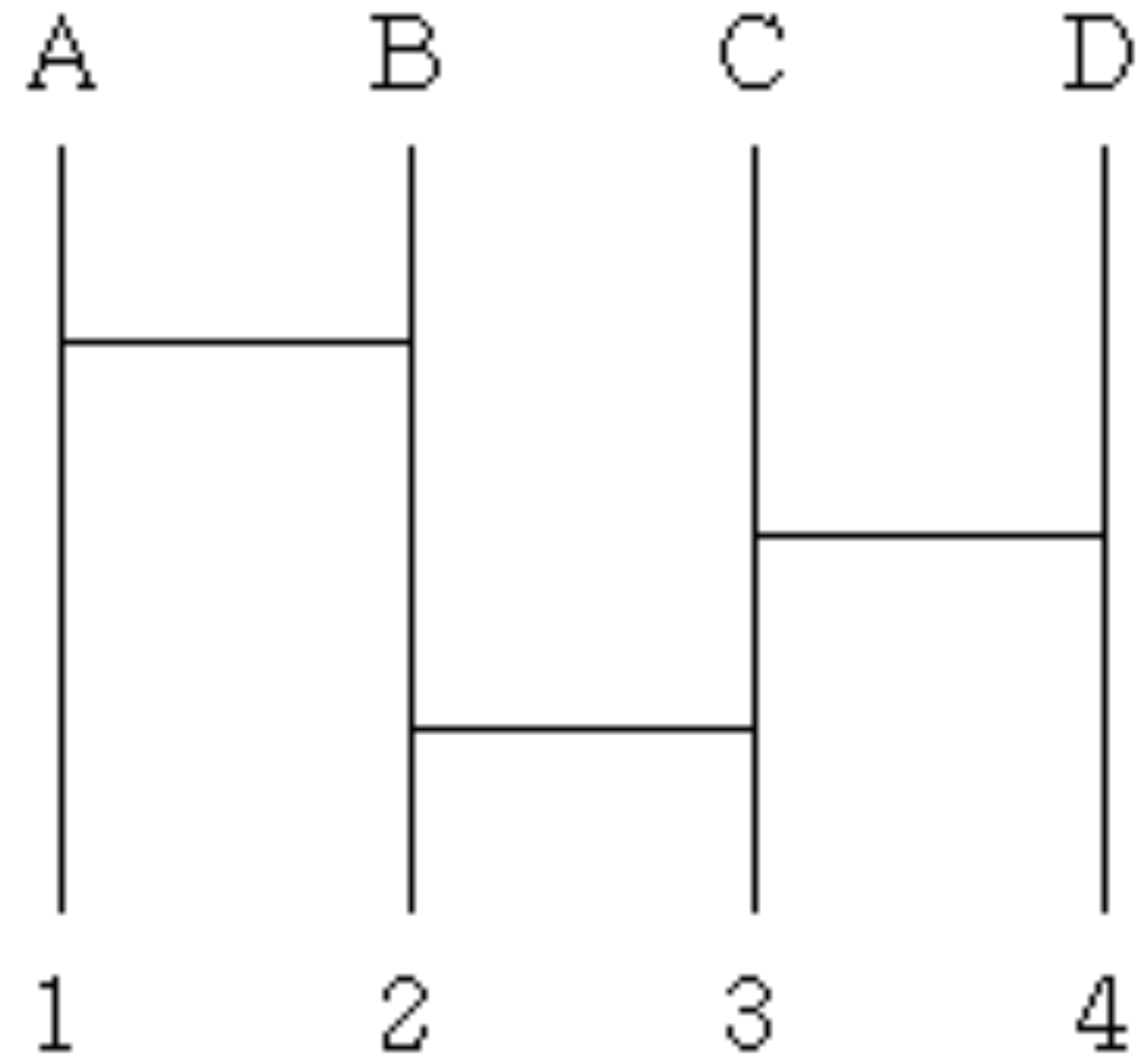


# 사다리 게임

26

<https://www.acmicpc.net/problem/2008>

- $D[i][j]$  =  $i$ 번째 가로선 까지 고려했을 때 start에서  $j$ 를 가는 최소 비용



# 사다리 게임

<https://www.acmicpc.net/problem/2008>

```
for (int i=0; i<m; i++) {  
    if (i == start) {  
        d[0][i] = 0;  
    } else {  
        d[0][i] = abs(start - i) * add;  
    }  
}
```

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를  $k$ 라고 했을 때
- $j == k$  인데, 가로선이  $j$ 나  $j+1$ 에 위치한 경우

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

```
if (k == j && (a[i] == k || a[i]+1 == k)) {  
    if (d[i][j] > d[i-1][k] + del) {  
        d[i][j] = d[i-1][k] + del;  
    }  
}
```

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를  $k$ 라고 했을 때
- 가로선이  $j$ 와  $k$  사이에 위치한 경우

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

```
else if ((k <= a[i] && a[i] <= j-1) || (j <= a[i] && a[i] <= k-1))  
{  
    if (d[i][j] > d[i-1][k] + (abs(k-j)-1)*add) {  
        d[i][j] = d[i-1][k] + (abs(k-j)-1)*add;  
    }  
}
```

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를  $k$ 라고 했을 때
- 가로선이  $j$ 와  $k$  사이에 위치하지 않은 경우



# 사다리 게임

<https://www.acmicpc.net/problem/2008>

```
else {  
    if (d[i][j] > d[i-1][k] + abs(k-j)*add) {  
        d[i][j] = d[i-1][k] + abs(k-j)*add;  
    }  
}
```

# 사다리 게임

<https://www.acmicpc.net/problem/2008>

- 소스: <http://codeplus.codes/5c8d7d57d736405e8019ccfc09dd3792>

# 같은 탑

<https://www.acmicpc.net/problem/1126>

- N개의 조각이 주어졌을 때, 두 개의 탑을 만든다  $N \leq 50$
- 이때, 두 탑의 높이를 같게 만드려고 한다.
- 가능한 탑의 높이 중 최대값을 구하는 문제
- 모든 조각의 높이의 합은 500,000을 넘지 않는다

# 같은 탑

<https://www.acmicpc.net/problem/1126>

- 모든 조각의 높이의 합은 500,000을 넘지 않는다
- 즉, 두 탑의 최대 높이는  $500,000/2 = 250,000$  이다.

# 같은 탑

<https://www.acmicpc.net/problem/1126>

- 각각의 조각에 대해서 다음과 같은 세 가지를 결정할 수 있다.
- 첫 번째 탑에 조각을 올려놓는다
- 두 번째 탑에 조각을 올려놓는다
- 조각을 탑 위에 올려놓지 않는다

# 같은 탑

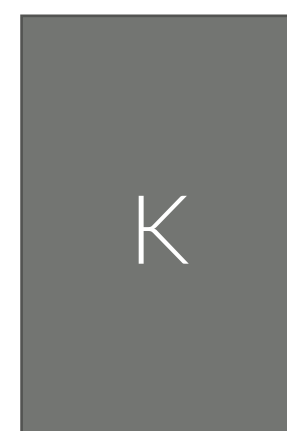
<https://www.acmicpc.net/problem/1126>

- 문제를 일반화 할 수 있다
- 탑 하나의 높이는 diff이고, 또 다른 탑의 높이는 0이다.
- 여기서, 조각을 적절히 놓아서 만들 수 있는 가장 큰 두 탑의 높이
- 이 때, 두 탑의 높이는 같아야 한다.
- $D[N][diff]$  = 조각이 N개 남았고, 높은 탑의 높이가 diff

# 같은 탑

<https://www.acmicpc.net/problem/1126>

- 탑 하나의 높이는  $\text{diff}$ 이고, 또 다른 탑의 높이는 0이다.
- 조각의 높이는  $K$ 이다.

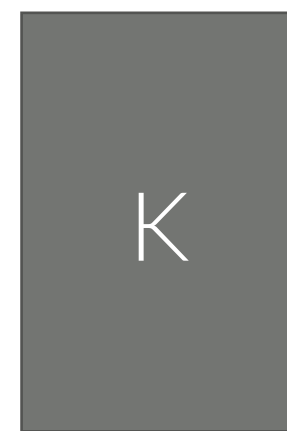


# 같은 탑

40

<https://www.acmicpc.net/problem/1126>

- 블록을 diff인 탑에 놓는 경우
- 블록을 0인 탑에 놓는 경우

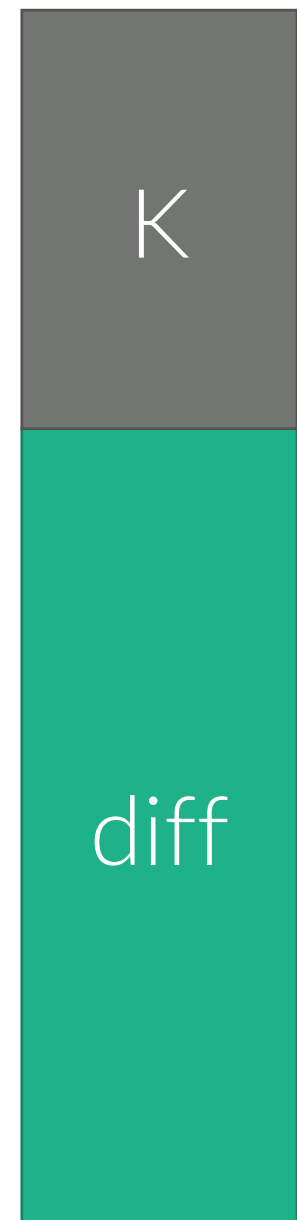




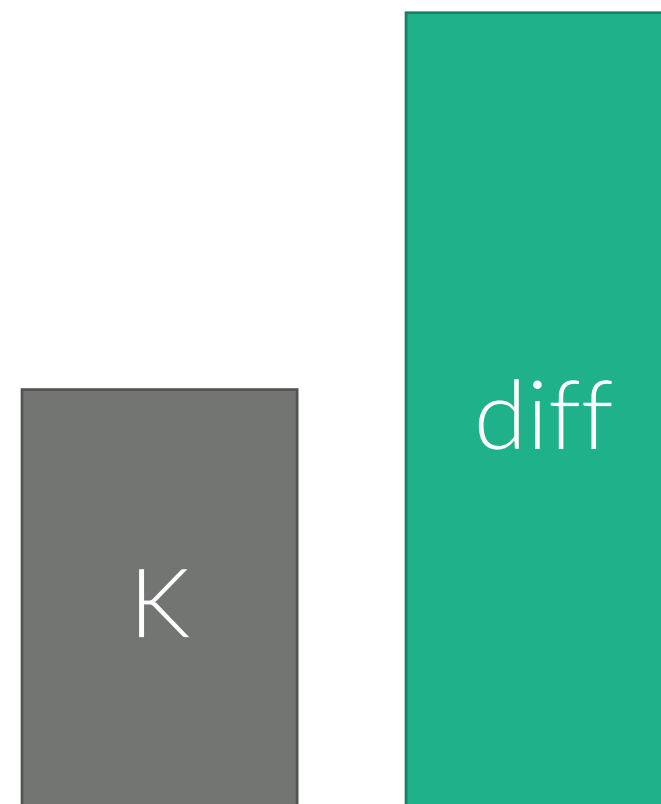
# 같은 탑

41

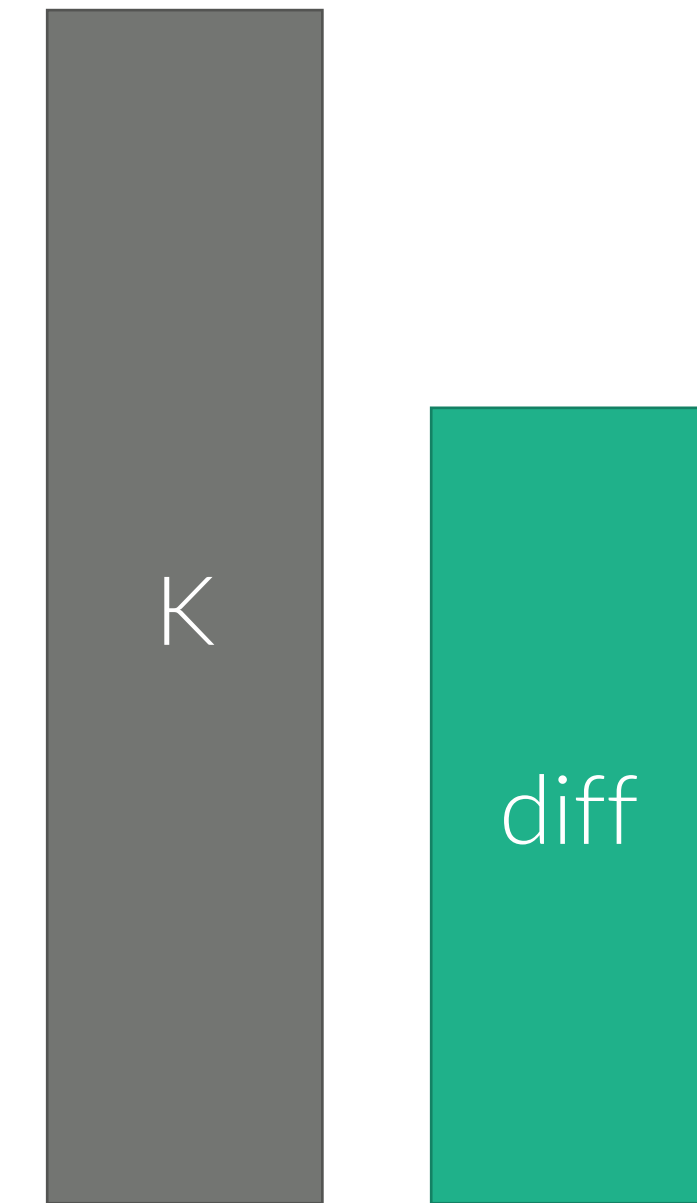
<https://www.acmicpc.net/problem/1126>



$D[K+1][diff+A[K]]$



$A[K] + D[K+1][diff-A[K]]$



$diff + D[K+1][A[K]-diff]$

# 같은 탑

42

<https://www.acmicpc.net/problem/1126>

- 소스: <http://codeplus.codes/98bcc4eb1bbc42449b41333ae18c2181>

# 경로 찾기

<https://www.acmicpc.net/problem/1513>

- $N \times M$  크기의 직사각형 도시에 오락실이  $C$ 개 있다.
- $(1, 1)$ 에서 출발해서  $(N, M)$ 에 도착하는 방법 중에서 오락실을 0개, 1개,  $\dots$ ,  $C$ 개 방문하고 도착하는 경우의 수를 구하는 문제
- 오락실 번호는 증가하는 순으로만 갈 수 있다.
- $(r, c)$ 에서 갈 수 있는 칸은  $(r+1, c)$ ,  $(r, c+1)$  이다.

# 경로 찾기

<https://www.acmicpc.net/problem/1513>

- $D[i][j][k][len]$  = (1, 1)에서 (i, j)에 오락실 len개를 방문해서도착하는 방법의 수 (마지막으로 방문한 오락실: k)

# 경로 찾기

<https://www.acmicpc.net/problem/1513>

- $D[i][j][k][len] = (1, 1)$ 에서  $(i, j)$ 에 오락실  $len$ 개를 거쳐서 도착하는 방법의 수 (마지막으로 방문한 오락실:  $k$ )
- $(i, j)$ 가 오락실인 경우
- $d[i-1][j][0][len-1] + d[i][j-1][0][len-1] + d[i-1][j][1][len-1] + d[i][j][1][len-1] + \dots + d[i-1][j][k-1][len-1] + d[i][j-1][k-1][len-1]$

# 경로 찾기

<https://www.acmicpc.net/problem/1513>

- $D[i][j][k][len]$  = (1, 1)에서 (i, j)에 오락실 len개를 거쳐서 도착하는 방법의 수 (마지막으로 방문한 오락실: k)
- (i, j)가 오락실이 아닌 경우
- $d[i-1][j][k][len] + d[i][j-1][k][len]$

# 경로 찾기

<https://www.acmicpc.net/problem/1513>

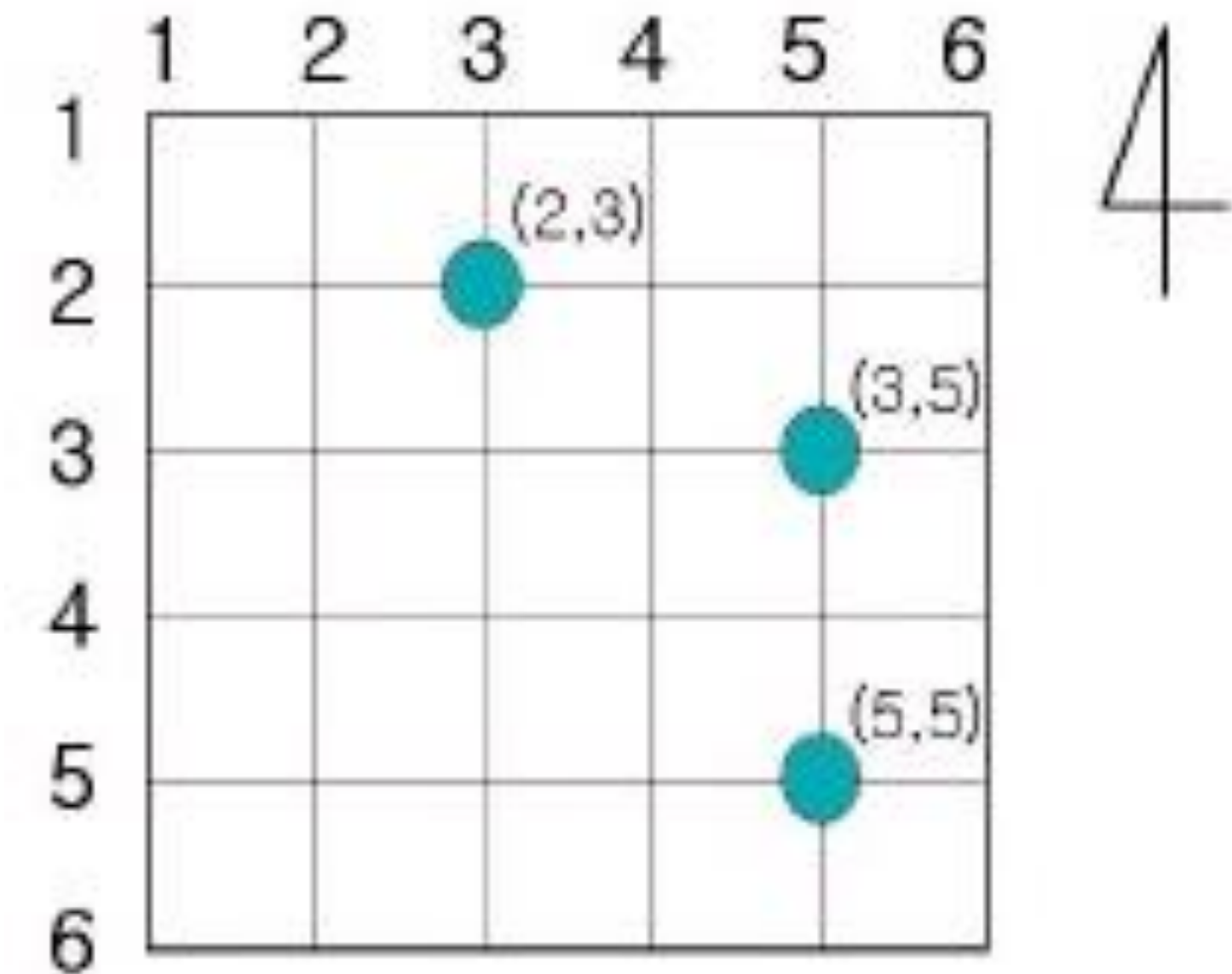
- 소스: <http://codeplus.codes/a85dfed2f7c0413da6c6a46d81b9f5b8>

# 경찰차

48

<https://www.acmicpc.net/problem/2618>

- N개의 동서방향 도로, 남북방향 도로가 있다.  $5 \leq N \leq 1,000$
- 경찰차 1은 (1, 1), 경찰차 2는 (N, N)에 있다.
- 사건이 일어나면 두 경찰차 중 하나가 이동해야 한다.
- 이동하는 거리의 최소값을 구하는 문제
- 사건의 발생 순서가 (3, 5), (5, 5), (2, 3) 라면 정답 = 9



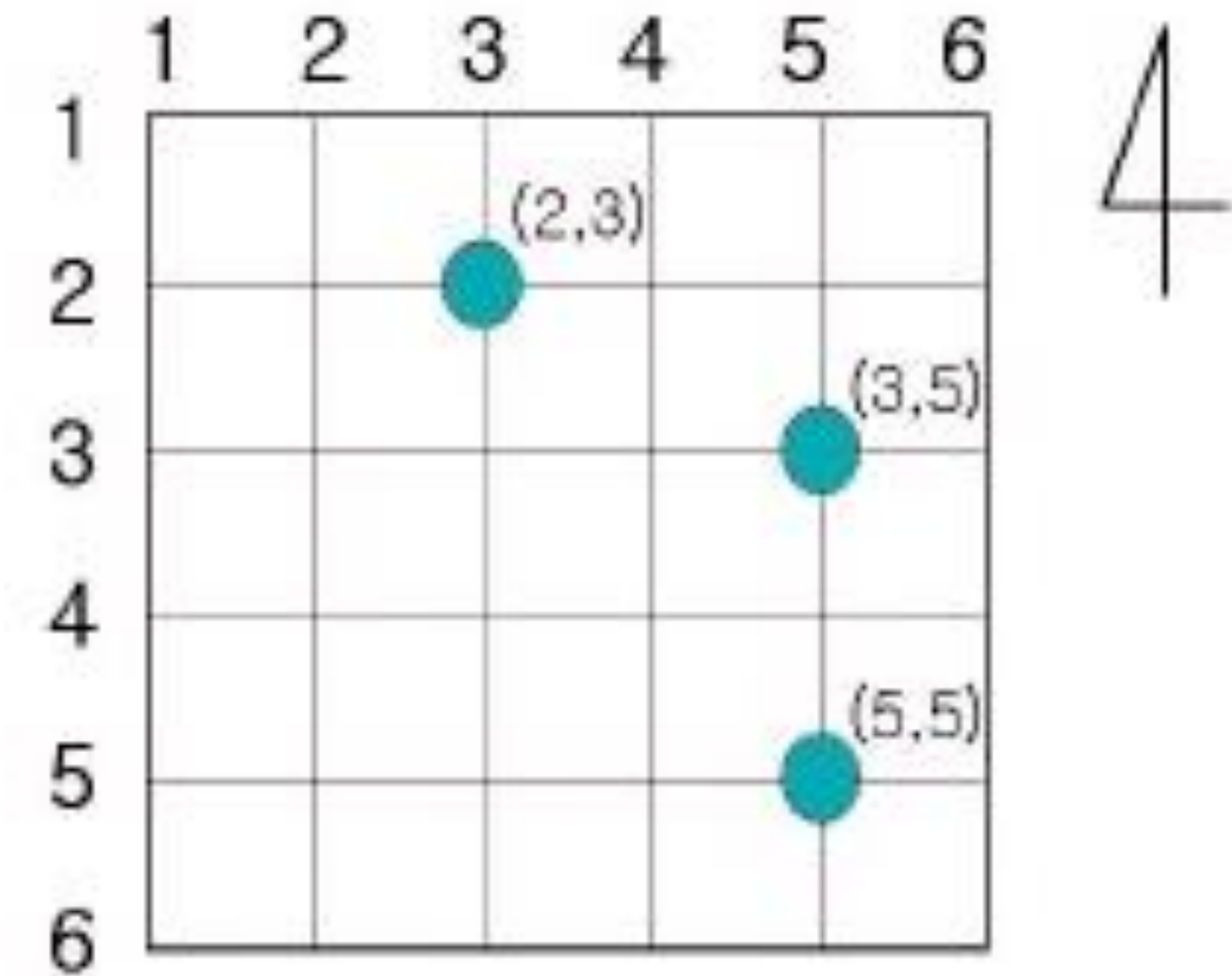


# 경찰차

49

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값

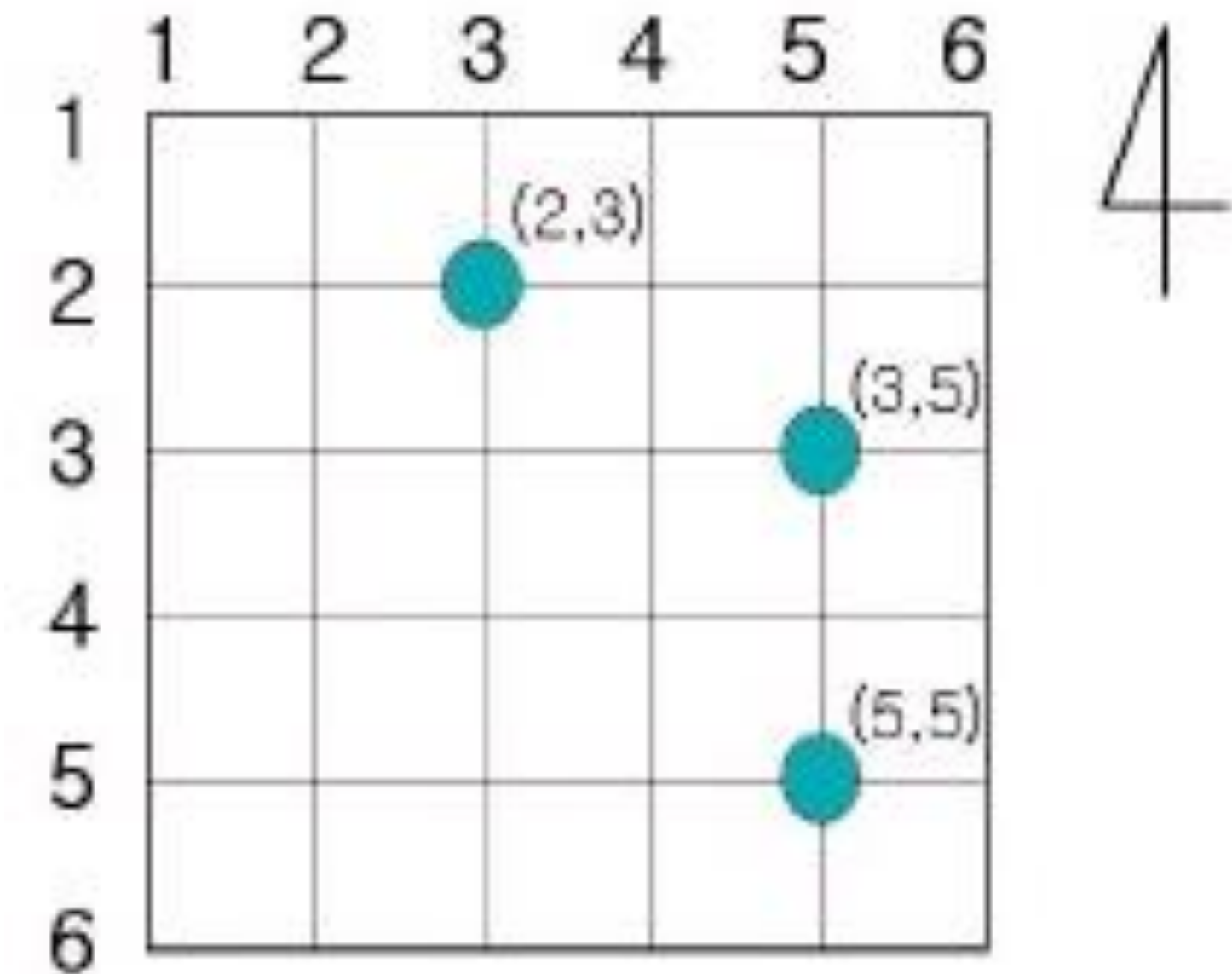


# 경찰차

50

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- $i$ 와  $j$ 중에서 최대값은 마지막으로 발생한 사건의 번호이다



# 경찰차

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- $i$ 와  $j$ 중에서 최대값은 마지막으로 발생한 사건의 번호이다
- 1번 경찰차가  $i$ 에 있을 때
  - 2번 경찰차가  $i-1$ 에 있는 경우
  - $i-1$ 이 아닌 다른 곳( $j$ )에 있는 경우로 나눌 수 있다.

# 경찰차

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- 1번 경찰차가  $i$ 에 있을 때
  1. 2번 경찰차가  $i-1$ 에 있는 경우 ( $D[i][i-1]$ )
    - 1번 경찰차는  $j$ 에 있어야 한다.
  2. 2번 경찰차가  $i-1$ 이 아닌 다른 곳 ( $j$ )에 있는 경우 ( $D[i][j]$ )
    - 1번 경찰차는 반드시  $i-1$ 에 있어야 한다.

# 경찰차

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- 1번 경찰차가  $i$ 에 있을 때
  1. 2번 경찰차가  $i-1$ 에 있는 경우 ( $D[i][i-1]$ )
    - 1번 경찰차는  $j$ 에 있어야 한다. ( $D[j][i-1]$ )
  2. 2번 경찰차가  $i-1$ 이 아닌 다른 곳 ( $j$ )에 있는 경우 ( $D[i][j]$ )
    - 1번 경찰차는 반드시  $i-1$ 에 있어야 한다. ( $D[i-1][j]$ )

# 경찰차

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- 1번 경찰차가  $i$ 에 있을 때
  1. 2번 경찰차가  $i-1$ 에 있는 경우 ( $D[i][i-1]$ )
    - 1번 경찰차는  $j$ 에 있어야 한다. ( $D[j][i-1] + \text{dist}[j][i]$ )
  2. 2번 경찰차가  $i-1$ 이 아닌 다른 곳 ( $j$ )에 있는 경우 ( $D[i][j]$ )
    - 1번 경찰차는 반드시  $i-1$ 에 있어야 한다. ( $D[i-1][j] + \text{dist}[i-1][i]$ )

# 경찰차

<https://www.acmicpc.net/problem/2618>

- $D[i][j]$  = 1번 경찰차가  $i$ , 2번 경찰차가  $j$ 에 있을 때, 이동하는 거리의 최소값
- 1번 경찰차가  $i$ 에 있을 때
  1. 2번 경찰차가  $i-1$ 에 있는 경우 ( $D[i][i-1]$ )
    - 1번 경찰차는  $j$ 에 있어야 한다. ( $D[j][i-1] + \text{dist}[j][i]$ )
  2. 2번 경찰차가  $i-1$ 이 아닌 다른 곳 ( $j$ )에 있는 경우 ( $D[i][j]$ )
    - 1번 경찰차는 반드시  $i-1$ 에 있어야 한다. ( $D[i-1][j] + \text{dist}[i-1][i]$ )
- 2번 경찰차에 대해서도 비슷한 처리를 하면 된다.

# 경찰차

56

<https://www.acmicpc.net/problem/2618>

- 소스: <http://codeplus.codes/bf154207d1c5421482817af7ab5e89c4>



# 직사각형 만들기

<https://www.acmicpc.net/problem/1801>

- 막대 N개를 가지고 있다. 이 막대를 이용해 만들 수 있는 가장 큰 직사각형의 넓이를 구하는 문제
- $4 \leq N \leq 16$ , 막대의 길이  $\leq 10$
- 가지고 있는 막대의 길이가 1, 3, 3, 4, 5, 7라면
- 3 두 개와 5, 그리고 1과 4를 붙여 5를 만들면  $3 \times 5$  크기의 직사각형을 만들 수 있다.

# 직사각형 만들기

58

<https://www.acmicpc.net/problem/1801>

- $D[i][L1][L2][L3][L4] = A[1] \sim A[i]$ 의 막대를 이용해서 L1, L2, L3, L4를 만들 수 있는가?

# 직사각형 만들기

<https://www.acmicpc.net/problem/1801>

- $D[i][L1][L2][L3][L4] = 1$
- if
- $D[i-1][L1][L2][L3][L4] == 1$  or
- $D[i-1][L1-A[i]][L2][L3][L4] == 1$  or
- $D[i-1][L1][L2-A[i]][L3][L4] == 1$  or
- $D[i-1][L1][L2][L3-A[i]][L4] == 1$  or
- $D[i-1][L1][L2][L3][L4-A[i]] == 1$

# 직사각형 만들기

60

<https://www.acmicpc.net/problem/1801>

- 필요한 공간의 크기
- $16 \times 80 \times 80 \times 80 \times 80$

# 직사각형 만들기

<https://www.acmicpc.net/problem/1801>

- 가장 앞의 i는 의미가 없다.
- $D[L1][L2][L3][L4] = L1, L2, L3, L4$ 를 만들 수 있는가?

# 직사각형 만들기

<https://www.acmicpc.net/problem/1801>

- $L1 == L2$ ,  $L3 == L4$ 인  $L1, L2, L3, L4$  중에서  $L1 \times L3$ 의 최대값이 정답

# 직사각형 만들기

<https://www.acmicpc.net/problem/1801>

- 소스 1: <http://codeplus.codes/0d9a3ab22a1e402bb9e8a6430ce57c86>
- 소스 2: <http://codeplus.codes/f058c4d68c604840bd2cda26258f405d>

# 서로소의 개수

<https://www.acmicpc.net/problem/1750>

- 크기가  $N$ 인 수열  $S$ 가 주어진다. 이 때, 한 개 이상을 선택했을 때, 선택한 수의 최대공약수가 1이 되는 것의 개수를 구하는 문제
- $1 \leq N \leq 100, 1 \leq S[i] \leq 100,000$



# 서로소의 개수

65

<https://www.acmicpc.net/problem/1750>

- $D[N][G]$  = N번째 수 까지 있을 때, 최대 공약수가 G가 되는 것의 개수

# 서로소의 개수

<https://www.acmicpc.net/problem/1750>

- $D[i][A[i]] = 1$
- $i$ 번째를 선택하지 않는 경우
- $i$ 번째를 선택하는 경우

# 서로소의 개수

<https://www.acmicpc.net/problem/1750>

- $D[i][A[i]] = 1$
- $i$ 번째를 선택하지 않는 경우
  - $D[i] += D[i-1][j]$
- $i$ 번째를 선택하는 경우
  - $D[i][\gcd(j, A[i])] += D[i-1][j]$

# 서로소의 개수

68

<https://www.acmicpc.net/problem/1750>

- 정답은?
- $D[n][1]$

# 서로소의 개수

<https://www.acmicpc.net/problem/1750>

- 소스: <http://codeplus.codes/d5b71896c024461aaddcc56b39bd109e>

# 그래프 매칭

<https://www.acmicpc.net/problem/3644>

- 사이클 그래프  $C_n$ 이 주어졌을 때, 매칭의 개수를 구하는 문제
- 매칭: 공통 정점을 갖지 않는 간선의 집합

# 그래프 매칭

<https://www.acmicpc.net/problem/3644>

- 먼저, 직선 그래프에서 문제를 풀어보고, 매칭의 개수를  $F_n$ 이라고 한다.
- $F_2 = 2$

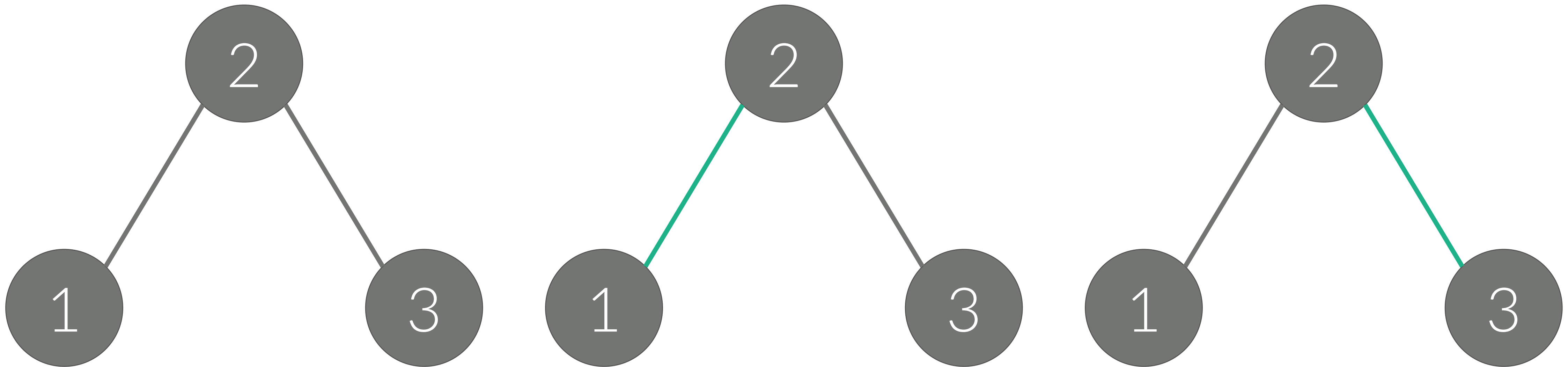


# 그래프 매칭

72

<https://www.acmicpc.net/problem/3644>

- 먼저, 직선 그래프에서 문제를 풀어보고, 매칭의 개수를  $F_n$ 이라고 한다.
- $F_3 = 3$





# 그래프 매칭

73

<https://www.acmicpc.net/problem/3644>

- $F_n = F_{n-1} + F_{n-2}$

# 그래프 매칭

<https://www.acmicpc.net/problem/3644>

- 사이클 그래프  $C_n$  에서 매칭의 개수 ( $L_n$ )를 구해보자
- $(n, 1)$  간선을 사용하지 않는 경우:  $F_n$
- $(n, 1)$  간선을 사용하는 경우:  $F_{n-2}$
- $L_n = F_n + F_{n-2} = L_{n-1} + L_{n-2}$
- $L_n = \text{Lucas Number}$

# 그래프 매칭

75

<https://www.acmicpc.net/problem/3644>

- 소스: <http://codeplus.codes/cc5339bd6e91474d9fda3f4b3319cd16>