

## Homework #3

Due: 11/22

### HW3 instructions

모든 제출 파일은 **학번\_문제번호.py** 로 작성해주시고, 하나의 zip 파일로 묶어서 제출해 주세요.  
 코드는 Python(.py extension)으로 작성하시기 바랍니다. 채점과 Input output 처리는 기존의 HW1, HW2 coding part 와 동일한 포맷으로 진행합니다. 이전 공지를 참고해주세요. **(Total 100 points)**

파일 1: 학번\_1.py

파일 2: 학번\_2.py

파일 3: 학번\_3.py

파일 4: 학번\_4.py

파일 5: 학번\_5.py

→ **HW3\_학번.zip** 압축 후 제출

주의: 알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.

만점을 받기 위해서는 runtime 을 최소화하는 방향으로 알고리즘을 디자인하는 것을 장려함.

유사도 검사 후 표절 적발 시 HW3 0점 처리.

### 1. 경유지 포함 최단 경로 탐색 (15 points)

달구는 오늘 안에 처리해야 할 일이 다음과 같다.

1. 학술정보관에 들러 빌린 책을 반납해야 한다.
2. 자전거 수리점에 자전거를 맡겨야 한다.

기숙사에서 출발하여 외부 헬스장으로 최단 거리로 이동하는 중에 위의 일들을 모두 처리하려 한다. 이 문제는 아래와 같이 그래프 문제로 바꿔볼 수 있다.

- a. 특정 장소를 node(vertex)로, 두 node 사이의 거리를 edge 로 하여 n 개의 node 로 구성된 undirected graph 상에 mapping 한다.
- b. 1 번 node 에서 n 번 node 로 최단 거리로 이동한다. 이 때, 특정 두 node 를 반드시 통과해야 한다는 제약이 있다.

주어진 두 경유지를 포함하는 shortest path 를 구하는 프로그램을 작성하시오.

Input Format

첫째 줄에는 node 의 개수  $n$  과 edge 의 개수  $e$  가 공백으로 구분되어서 주어진다.

각 node 의 이름은 1 부터  $n$  까지의 정수로 매핑한다. ( $2 \leq n \leq 400, 0 \leq e \leq 100,000$ )

두번째 줄에는 서로 다른 두 경유지의 node 번호가 공백으로 구분되어서 주어진다.

그 아래  $e$  개의 줄에는 각 edge 의 정보가 담겨있는데,  $s$   $d$   $w$  가 공백으로 구분되어서 주어진다.

( $s$ : 출발 node 의 번호,  $d$ : 도착 node 번호,  $w$ : edge [ $s$ -  $d$ ]의 weight) ( $1 \leq w \leq 1,000$ )

방문했던 node 와 edge 를 다시 방문할 수 있다.

임의의 두 node 사이 edge 는 0 개 또는 1 개이다.

Output Format

shortest path 의 weight 합을 출력한다. 경로가 존재하지 않으면 “NO WAY”를 출력한다.

Sample Input

```
5 8
2 4
1 2 1
2 3 2
3 4 3
4 5 4
1 3 5
4 1 1
2 5 2
3 5 3
```

Sample Output

```
5
```

(알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.)

**2. 달구의 세계 일주 – 여행 자금 마련 (20 points)**

세계 일주는 달구의 로망이다. 코딩 실력이 출중한 달구는 코딩 알바를 뛰여 여행 자금을 마련하려고 한다.  $(n + 1)$ 일 뒤에 세계 일주를 떠나기 위해, 그동안 최대한 많은 돈을 벌어드는 것이 달구의 목표다.

일하려는 회사에서는 매일 하나의 신규 프로젝트를 런칭한다. 각각의 프로젝트는 풀타임으로 일했을 때 걸리는 기간  $D_i$ 와 완료했을 때 보수  $P_i$ 로 표현할 수 있다.

( $D_i$ 일) ( $P_i$ 만원)

예를 들어,  $n = 10$  이고, 아래와 같은 프로젝트 런칭 계획표가 주어졌다고 생각해보자.

$n = 10$	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
$D_i$	4	3	1	1	3	5	3	1	4	2
$P_i$	10	20	10	20	15	50	20	3	40	100

Day 1 에 런칭된 프로젝트는 총 4 일이 걸리며, 완료했을 때 보수는 10 만원이다.

이 때, 걸리는 기간은 풀타임 기준이기 때문에 근로기준법 상 하루에 여러 프로젝트에 참여할 수는 없다. 예를 들어, Day 1 에 런칭된 프로젝트에 참여한다면 Day 2, 3, 4 에 런칭되는 프로젝트에는 참여할 수 없다. Day 6 에 런칭된 프로젝트에 참여한다면 Day 7, 8, 9, 10 에 런칭되는 프로젝트에는 참여할 수 없다.

또한, Day 11(Day  $n+1$ )에는 세계 일주를 떠나야 하기에, Day 9, 10 에 런칭되는 프로젝트는 참여할 수 없다.

위 예제에서 세계 일주 전에 할 수 있는 코딩 알바의 최대 보수는 Day3, Day 4, Day 6 에 런칭되는 프로젝트에 참여하는 것이며  $10 + 20 + 50 = 80$  만원이다.

코딩 알바를 통해 얻을 수 있는 최대 보수를 구하는 프로그램을 작성하시오.

Input Format

첫째 줄에  $n$  이 주어진다. ( $1 \leq n \leq 100,000$ )

그 아래  $n$  개의 줄에  $D_i$ 와  $P_i$ 가 공백으로 구분되어서 주어지며, Day 1 부터  $n$  까지 순서대로 주어진다. ( $1 \leq D_i \leq 50, 1 \leq P_i \leq 800$ )

Output Format

첫째 줄에 달구가 코딩 알바를 통해 얻을 수 있는 최대 이익을 출력한다.

Sample Input

```
10
4 10
3 20
1 10
1 20
3 15
5 50
3 20
1 3
4 40
2 100
```

Sample Output

```
80
```

DP 추천

(알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.)

### 3. 달구의 세계 일주 – 전체 동선 (20 points)

달구는 세계 일주를 떠나기 전에 대략적인 동선을 짜려고 한다.

이 때, 달구는 꼭 방문하고 싶은 국가들과 방문 순서를 정해 두었다.

같은 국가를 여러 번 방문할 수도 있다.

그런데, 임의의 두 국가 사이에는 직통 항공편이 있거나 다른 국가를 경유한 항공편이 있거나, 갈 방법이 전혀 없을 수 있다.

예를 들어 국가가 5 개 있고, 1-2 (2-1), 2-3 (3-2), 1-4 (4-1), 2-4 (4-2), 5-1 (1-5)의 항공편이 있고, 달구의 방문 계획이 **53234** 라면 **5-1-2-3-2-3-2-4** 을 통해 가능함을 알 수 있다.

괄호는  $i-j$  항공편이 있다면,  $j-i$  항공편도 있음을 뜻한다.

국가의 수( $N$ )와 국가 간 항공편 정보( $N \times N$ )가 주어져 있고, 달구가 꼭 방문하고자 하는  $M$  개의 국가(중복 가능)가 순서대로 주어졌을 때, 가능 여부를 판별하는 프로그램을 작성하시오.

Input Format

첫째 줄에 국가의 수,  $N$  이 주어진다. ( $N \leq 200$ ) 각 국가의 이름은 1 부터  $N$  까지의 정수로 매핑한다.

그 아래의  $N$  개의 줄들은  $N$  개의 0 또는 1 의 정수가 **공백 구분 없이** 주어진다.  $i$  번째 줄의  $j$  번째 수는  $i$  번 도시와  $j$  번 국가의 항공편 유무를 뜻한다. 1 이면 항공편이 존재한다는 뜻이고, 자기 자신으로의 항공편은 항상 0 으로 주어진다.  $i$  번째 줄  $j$  번째 수가 1 이라면  $j$  번째 줄  $i$  번째 수도 1 이다.

다음 줄 ( $N+2$  번째 줄)에는 꼭 방문하고자 하는 국가의 수  $M$  이 주어진다.

중복 방문 국가는 방문 횟수만큼 더한다. ( $M \leq 1,000$ )

마지막 줄에는 꼭 방문하고자 하는  $M$  개의 국가들이 **공백으로 구분하여** 순서대로 주어진다.

방문했던 국가를 다시 방문할 수 있다.

임의의 두 국가 사이 항공편은 0 개 또는 1 개이다.

Output Format

가능하면 YES, 불가능하면 NO 를 출력한다.

Sample Input

```
5
01011
10110
01000
11000
10000
5
5 3 2 3 4
```

Sample Output

```
YES
```

(알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.)

## 4. 달구의 세계 일주 - 국가별 동선 (25 points)

달구는 세계 일주를 떠나기 전에 마지막으로 국가별 동선을 짜려고 한다.

달구는 방문할 국가별 등고선 정보를 수치화한 지도들을 구했다.

이 지도들은 항상 직사각형 ( $M \times N$  matrix) 이라고 가정하자.

예를 들어, 그 지도 중 하나는 아래 그림과 같다. ( $4 \times 5$  matrix)

각 칸에는 해당 지역의 높이가 적혀 있다.

특정 칸에서 다른 칸으로의 이동은 상하좌우로 인접한 칸으로만 가능하다.

100	90	70	50	40
60	100	80	10	15
40	40	15	7	30
20	14	12	5	1

달구는 항상 지도 상 가장 왼쪽 윗 칸의 공항에서 내려 출발한다. 달구의 목표는 지도 상 가장 오른쪽 아래 칸으로 이동하는 것이다. 이 때, 달구는 항상 높이가 더 낮은 칸으로만 이동하여 목적지까지 가려고 한다. 이 예시에 대해서는 아래 세 동선이 가능하다.

100	90	70	50	40
60	100	80	10	15
40	40	15	7	30
20	14	12	5	1

100	90	70	50	40
60	100	80	10	15
40	40	15	7	30
20	14	12	5	1

100	90	70	50	40
60	100	80	10	15
40	40	15	7	30
20	14	12	5	1

상하좌우로 인접한 칸을 통해서만 이동 가능하기에 아래 동선은 불가능하다.

100	90	70	50	40
60	100	80	10	15
40	40	15	7	30
20	14	12	5	1

등고선 지도가 input 으로 들어왔을 때, 높이가 더 낮은 칸으로만 이동하여 가장 왼쪽 윗 칸에서 출발해서 가장 오른쪽 아래 칸에 도착하는 동선 개수를 구하는 프로그램을 작성하시오.

Input Format

$M \times N$  matrix 에 대해, 첫째 줄에  $M$  과  $N$  이 공백으로 구분되어서 주어진다.

( $M$  과  $N$  은 500 이하의 자연수)

그 아래  $M$  개의 줄에 해당 행의 각  $N$  개 칸의 높이가 공백으로 구분되어서 주어진다.

(높이는 10000 이하의 자연수)

Output Format

첫째 줄에 가능한 동선의 수를 출력한다.

Sample Input

```
4 5
100 90 70 50 40
60 100 80 10 15
40 40 15 7 30
20 14 12 5 1
```

Sample Output

3

(알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.)

DP 추천



**5. 달구의 세계 일주 – 치즈 축제 (20 points)**

네덜란드에 도착한 달구는 치즈 축제에 갔다.

부스마다 각기 다른 종류의 치즈들이 놓여있고, 부스들은 일렬로 놓여 있다.

축제 운영방침은 다음과 같다.

- 1. 모두에게 무료 시식 기회를 제공함.*
- 2. 최대한 많은 사람들이 다양한 치즈를 시식할 기회를 제공하기 위해 연속으로 인접한 세 개의 부스에 방문하는 것을 금함.*

$N$  개의 부스에 대하여 각 부스에서 제공하는 치즈의 만족도가 수치화되어 주어진다고 가정하자. 달구는 이 곳에서 만족도를 최대화하고자 한다.

예를 들어 8 개의 부스가 있고, 각각의 부스에서 먹을 수 있는 치즈의 만족도가 10, 5, 1, 14, 7, 6, 1, 15 이라면, 1<sup>st</sup>, 2<sup>nd</sup>, 4<sup>th</sup>, 5<sup>th</sup>, 7<sup>th</sup> 8<sup>th</sup> 부스를 방문하였을 때 총 만족도가  $10 + 5 + 14 + 7 + 1 + 15 = 52$  으로 최대이다.

이 때, 달구가 가장 높은 만족도를 얻을 수 있도록 하는 프로그램을 작성하시오.

Input Format

첫째 줄에 부스의 개수  $N$ 이 주어진다. ( $1 \leq N \leq 10,000$ )

그 아래  $N$  개의 줄에 각 부스에서 먹을 수 있는 치즈의 만족도가 순서대로 주어진다.

치즈의 만족도는 0 이상 1000 이하의 정수다. ( $0 \leq \text{치즈의 만족도} \leq 1,000$ )

Output Format

첫째 줄에 가능한 가장 높은 만족도를 출력한다.

Sample Input

8  
10  
5  
1  
14  
7  
6  
1  
15

Sample Output

52

DP 추천

(알고리즘의 runtime 이 길면 특정 testcase 들의 경우 실행시간 초과로 오답이 될 수 있음.)