

# [Python트랙] 월말평가2 - 알고리즘 기본



## | Background

- ✓ 선형 자료에 대한 이해와 활용
- ✓ 탐색에 대한 이해와 활용
- ✓ 백트래킹에 대한 이해와 활용

## | Goal

- ✓ 선형 자료구조를 활용한 알고리즘을 이해하고 문제를 해결한다.
- ✓ 탐색을 통해 문제를 해결한다.
- ✓ 백트래킹을 적용해 효율적으로 문제를 해결할 수 있다.

## | 환경 설정

1) Pycharm(Python3.7이상)을 이용해서 코드를 작성하고 결과를 확인한다.

- 새로운 Pycharm 프로젝트를 생성 후 코드를 작성한다.

2) 파일 이름 및 제출 방법

- 1, 2번 문제에 대한 소스 파일은 Algo문제번호\_지역\_반\_이름.py로 만든다.

- pypy의 경우 폴더, 프로젝트, 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 변경한다.

- 3번은 텍스트 파일로 작성한다.

Algo1\_서울\_1반\_이싸피.py

Algo2\_서울\_1반\_이싸피.py

Algo3\_서울\_1반\_이싸피.txt

- 위 3개의 파일만 지역\_반\_이름.zip으로 압축하여 제출한다.

서울\_1반\_이싸피.zip

(탐색기에서 파일 선택 후 오른쪽 클릭 - 보내기 - 압축(zip)폴더 선택)

3) 채점

- 주석이 없는 경우, 주석이 코드 내용과 맞지 않는 경우, 지정된 출력 형식을 만족하지 않는 경우 해당 문제는 0점 처리될 수 있다.

- import를 사용한 경우 해당 문제는 0점 처리될 수 있다. (import sys도 예외 없음)

4) 테스트케이스는 부분적으로 제공되며, 전체가 공개되지는 않는다.

5) 각 문제의 배점이 다르므로 표기된 배점을 반드시 확인한다.

- 1번 40점, 2번 35점, 3번 25점

## 성실과 신뢰로 테스트에 임할 것 (부정 행위시 강력 조치 및 근거가 남음)

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시, 0점 처리 및 학사 기준에 의거 조치 실시 예정

# [Python트랙] 월말평가2 - 알고리즘 기본



## | 문제1 : 미생물 이동 ( 배점 : 40점 )

김싸피가 실험중인 미생물은 N개의 활동 구역을 왼쪽에서 오른쪽으로 이동하고 있다. 미생물은 어떤 구역에서 먹이를 먹으면 오른쪽으로 K칸 만큼 더 이동할 수 있고, 추가로 먹이를 먹지 못하면, K번째 칸에 멈추게 된다고 한다.



먹이가 있는 활동구역에 대한 정보가 주어지고, 미생물이 가장 왼쪽 칸에 놓여지면, **미생물이 최대 몇 번째 칸까지 이동할 수 있는지** 알아내는 프로그램을 만드시오.

먹이가 있으면 1, 없으면 0으로 표시된 N개의 활동 구역과 미생물이 먹이를 먹고 이동하는 최대 칸 수 K가 제공된다. 첫 번째 구역에는 항상 먹이가 있다.

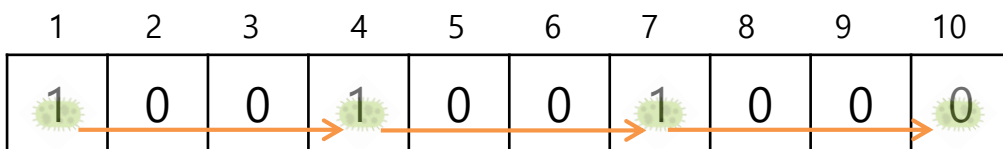


그림1. N = 10, K = 3인 경우

그림1의 경우, N = 10, K = 3인 예로, 미생물이 먹이를 먹을 때마다 3칸씩 이동할 수 있으므로 이 경우 미생물은 10번째 칸까지 이동할 수 있다.

(다음 페이지에 계속)

# [Python트랙] 월말평가2 - 알고리즘 기본



1	0	0	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

그림2.  $N = 10$ ,  $K = 3$ 인 경우

먹이 배치가 그림2와 같은 경우, 9번째 칸의 먹이를 먹어도 활동구역을 벗어날 수는 없고 10번째 칸에 멈추게 된다. 따라서 최대 이동 위치는 10번째 구역이다.

1	0	0	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

그림3.  $N = 10$ ,  $K = 2$ 인 경우

그림3의 경우, 첫 번째 칸에서 먹이를 먹은 미생물은 3번째 칸에서 멈추게 된다.

1	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

그림4.  $N = 10$ ,  $K = 3$ 인 경우

그림4의 경우, 미생물은 5번째 칸에서 멈추게 된다.

(다음 페이지에 계속)

# [Python트랙] 월말평가2 - 알고리즘 기본



## [입력]

첫 줄에 활동 구역의 수 T가 주어지고, 활동 구역별로 첫 줄에 N과 K, 다음 줄에 N개 구역의 먹이 정보가 빈칸으로 구분되어 주어진다.

( $3 \leq T \leq 10$ ,  $1 \leq N$ ,  $K \leq 10$ )

## [출력]

각 활동 구역별로 한 줄에 #과 1번 부터인 구역번호, 빈칸에 이어 최대 몇 번째 구역에 이동할 수 있는지 출력한다.

### [입력 예시]

```
3
10 3
1 0 0 1 0 0 1 0 0 0
10 3
1 0 0 1 0 0 1 0 1 0
10 2
1 0 0 1 0 0 1 0 0 0
```

(algo1\_sample\_in.txt 참고)

### [출력 예시]

```
#1 10
#2 10
#3 3
```

(algo1\_sample\_out.txt 참고)

# [Python트랙] 월말평가2 - 알고리즘 기본



## | 문제2 : 외계인 자리 배정 (배점 : 35점)

- **import는 사용할 수 없습니다.**

이싸피는 외계인 담당 조직인 맨인블루의 요원이다. 이싸피의 오늘 업무는 입국심사를 기다리는 N명의 외계인에 대해 줄을 세우는 것이다. 외계인은 각자 고향에 따라 고유한 물질 구성을 갖고 있어 줄을 서는 순서에 따라 가스발생 위험도가 높아진다고 한다. 다행히 맨인블루의 데이터베이스에는 현재 대기중인 외계인끼리의 위험도가 다음과 같이 기록되어 있다.

위험도	A	B	C	뒤
A	0	7	2	
B	6	0	5	
C	4	3	0	
앞				

표1. 외계인이 차례로 서있을 때의 위험도 예

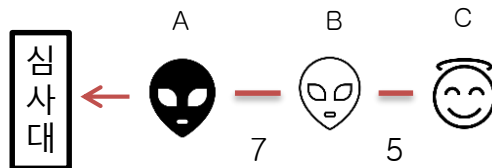


그림 1. 줄을 선 외계인의 예

표1에 의하면 A-B 순서로 줄을 서면 위험도 7, B-A 순서로 줄을 서면 위험도는 6 이다. 만약 A-B-C 순서로 줄을 서면 위험도는 A-B와 B-C에 의해 7+5인 12가 된다. 만약 C-B-A순서로 줄을 선다면 위험도는 3+6인 9가 된다. 이 표를 이용하면 A, B, C 외계인이 줄을 서는 순서를 잘 조절해 위험도를 최소로 줄일 수 있다. 이싸피가 줄을 잘 세웠을 때의 최소 위험도를 출력해보자.

# [Python트랙] 월말평가2 - 알고리즘 기본



앞에서 주어진 경우에는 A-C-B의 순서로 서면 위험도가 5로 가장 낮다.

## [입력]

첫 줄에 테스트 케이스 개수  $T$ 가 주어진다.

각 테스트 케이스의 첫 줄에는 외계인의 수  $N$ 이 주어진다. (줄을 서는  $N$ 명의 외계인은 모두 고향이 다르다.)

다음  $N$ 개의 줄에 걸쳐,  $N$ 개씩의 **자연수로 표시된 위험도  $A_{ij}$** 가 띄어쓰기로 구분되어 주어진다.

( $3 \leq N \leq 8$ ,  $1 \leq A_{ij} \leq 100$ ,  $1 \leq T \leq 10$ )

## [출력]

각 줄에 테스트케이스 번호를 #번호 형태로 출력하고, 공백에 이어 최소 위험도를 출력한다.

### [입력 예시]

```
1
3
0 7 2
6 0 5
4 3 0
```

(algo2\_sample\_in.txt 참고)

### [출력 예시]

```
#1 5
```

(algo2\_sample\_out.txt 참고)

# [Python트랙] 월말평가2 - 알고리즘 기본



## | 문제3 : 서술형 (배점 : 25점)

3-1. 스택과 큐의 특징에 대해 삽입과 삭제 순서를 중심으로 간단히 설명하시오.

3-2. DFS와 백트래킹의 차이에 대해 간단히 설명하시오.