

# [Python트랙] 과목평가5 – 알고리즘 응용



## | Background

- ✓ 배열에 대한 이해와 활용
- ✓ 완전탐색에 대한 이해와 활용
- ✓ 백트래킹에 대한 이해와 활용

## | Goal

- ✓ 반복문과 조건문을 이용하여 배열의 요소에 접근할 수 있다.
- ✓ 완전탐색으로 문제를 해결할 수 있다.
- ✓ 백트래킹을 적용해 문제를 해결할 수 있다.

## | 환경 설정

1) Pycharm과 pypy 또는 python을 이용해서 코드를 작성하고 결과를 확인한다. **새로운 프로젝트를 생성하지 않고 기존 프로젝트 사용시 부정행위로 간주 함.**

2) 파일 이름 및 제출 방법

- 1, 2번 문제에 대한 소스 파일 이름은 다음과 같이 영문으로 작성한다.

**서울 1반 이싸피라면, algo문제번호\_반\_이름.py 순서로 영문으로 작성**

**algo1\_01\_leessafy.py**

**algo2\_01\_leessafy.py**

- 3번 문제에 대한 답안 파일 이름은 .txt 형식으로 다음과 같이 영문으로 작성한다.

**algo3\_01\_leessafy.txt**

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

**서울\_1반\_이싸피.zip**

(탐색기에서 파일 선택 후 오른쪽 클릭 – 압축대상 – Zip 선택)

3) 채점

- 문제별로 부분 점수가 부여된다.
- 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] 과목평가5 - 알고리즘 응용



## | 문제1 : 단순 증가 패턴 ( 배점 : 40점 )

**import 사용금지.**

데이터 분석가 지망생인 김싸피는 숫자 뒤에 숨겨진 패턴을 찾는 것을 매우 좋아한다. 김싸피는 길이  $N$ 의 숫자열을 얻게 되었다. 김싸피는 이 수열에서 '단순증가패턴'을 찾으려고 한다. '단순증가패턴'이란, 패턴내 숫자 들이 **감소나 정체 없이 꾸준히 증가**하는 패턴이다. 또 김싸피는 패턴을 쉽게 파악하기 위해 수열을 길이  $M$ 인 윈도우로 나누어 윈도우 내에서 패턴을 찾아 보기로 했다. '단순증가패턴'을 가지는 윈도우는 몇 개인지 출력하는 프로그램을 작성하시오.

다음 예시는  $N = 9, M = 3$  인 예시이다.

1	2	4	4	6	3	2	4	9
---	---	---	---	---	---	---	---	---

총 3개의 윈도우를 만들 수 있고, 이 때 '1 2 4', '4 6 3', '2 4 9' 로 나누어 지므로 '단순증가패턴'은 2개이다.

$N$ 이  $M$ 의 배수가 아닌 경우 마지막 윈도우는 남은 원소에 대해서만 증가패턴을 확인한다. 단, 윈도우 내의 숫자가 1개인 경우 단순증가패턴으로 간주한다.

# [Python트랙] 과목평가5 - 알고리즘 응용



## [입력]

첫 번째 줄에 테스트케이스의 개수 T가 주어진다.

각 테스트 케이스의 첫 번째 줄에 N, M이 띄어쓰기로 구분되어 주어진다.

다음에 **N개의 정수**  $A_i$ 가 띄어쓰기로 구분되어 주어진다.

$(1 \leq M \leq N \leq 20, |A_i| \leq 100)$ ,

## [출력]

각 테스트케이스마다 '#tc'(tc는 테스트케이스 번호)를 출력하고, 단순증가 패턴을 가지는 윈도우의 개수를 출력한다.

### [입력 예시]

```
3
9 3
1 2 4 4 6 3 2 4 9
7 3
1 2 3 4 5 6 7
12 4
10 20 20 30 1 2 3 4 9 8 7
6
```

(algo1\_sample\_in.txt 참고)

### [출력 예시]

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

(algo1\_sample\_out.txt 참고)

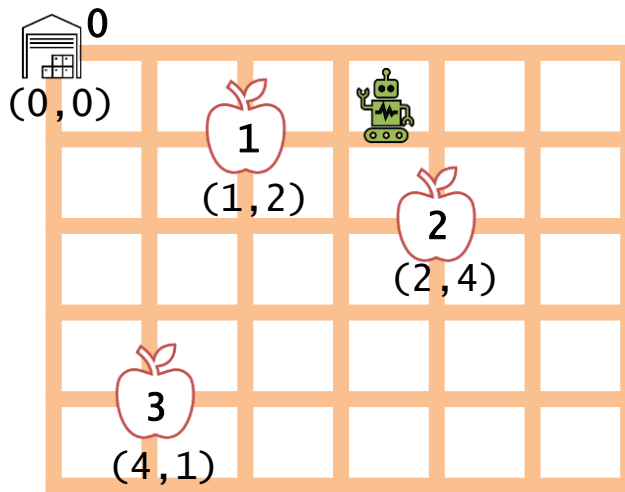
# [Python트랙] 과목평가5 - 알고리즘 응용



## | 문제2 : 사과 수확 (배점 : 35점)

**import 사용금지. 실행제한 시간 30초(10개의 테스트케이스에 대해 30초까지만 채점이 진행됨)**

싸피 농장에서는 새로운 로봇을 도입해 사과를 수확하기로 했다.



창고와 수확할 사과의 위치는 (x, y)로 표시되며, 로봇이 두 지점 (x1, y1), (x2, y2) 사이를 이동했을 때의 거리 d는 다음과 같이 계산한다.

$$d = |x1 - x2| + |y1 - y2|$$

로봇은 창고(0, 0)에서 출발 후 N개의 사과를 수확하고 다시 (0, 0)으로 돌아와야 한다.

사과를 수확하는 순서에 따라 로봇의 이동거리는 달라질 수 있다. 로봇이 사과를 수확하는 순서를 결정해 최소한의 거리만 이동할 수 있도록 하고, 이 거리를 출력하라.

# [Python트랙] 과목평가5 - 알고리즘 응용



앞의 그림의 경우 1 2 3 순으로 사과를 수확하면 이동거리가 16으로 최소 이동거리가 된다. (이동 거리가 최소인 수확 순서는 여러 가지가 있을 수 있다.)

## [입력]

첫 줄에 테스트케이스 수 T, 다음 줄부터 케이스별로 첫 줄에 N, 다음 줄부터 N줄에 걸쳐 사과의 위치 x, y가 빈칸으로 구분되어 주어진다. 사과의 위치가 (0, 0)인 경우는 없으며 사과의 위치는 모두 다르다.

(  $3 \leq T \leq 10$ ,  $0 \leq x, y \leq 100$ ,  $3 \leq N \leq 10$ )

## [출력]

#과 테스트케이스 번호, 빈칸에 이어 최소 이동거리를 출력한다.

### [입력 예시]

```
3
3
1 2
2 4
4 1
3
1 2
2 6
4 1
5
3 97
12 34
56 89
22 7
91 45
```

(algo2\_sample\_in.txt 참고)

### [출력 예시]

```
#1 16
#2 20
#3 394
```

(algo2\_sample\_out.txt 참고)

# [Python트랙] 과목평가5 - 알고리즘 응용



## | 문제3 : 컴퓨팅 사고력 ( 배점 : 25점 )

첫 페이지 '파일 이름 및 제출 방법'을 확인할 것

- (1) 정수  $n$ 에 대한 다음 명제에 대해 대우를 쓰고, 대우를 이용하여 명제를 증명하시오.

**명제:**  $3n + 5$ 가 홀수이면  $n$ 은 짝수이다.

- (1) 입력의 크기  $n$ 에 대한 어떤 코드의 연산량이 다음과 같은 때, 빅오 표기법으로 복잡도를 나타내시오. (풀이과정을 포함해 답을 적어야 함)

$$T(n) = T(n - 1) + 1, T(0) = 1$$