

[Python 트랙] 6회차 – 알고리즘



| Background

- ✓ 16진법과 2진법, 비트
- ✓ 계산 복잡도
- ✓ 순열, 조합, 부분집합

| Goal

- ✓ 진법 변환을 이해하고 활용할 수 있다.
- ✓ 순열을 이용한 완전탐색을 이해하고 활용할 수 있다.
- ✓ 계산 복잡도를 수식으로 표시하고 설명할 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 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 아래와 같이 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 50점, 2번 30점, 3번 20점

성실과 신뢰로 테스트에 볼 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

[Python 트랙] 6회차 – 알고리즘



| 문제 1 : 연속한 1의 개수 찾기 (배점 : 50점)

16진수를 2진수로 바꿨을 때, 연속한 1의 최대 개수를 출력 하는 프로그램을 작성하라.
16진수 한자리는 4비트 2진수로 변환한다.

16진수 E6D의 각 자리를 4비트 2진수로 변경하면 다음과 같다.

111001101101

이때 연속하는 1은 111, 11, 11, 1로 최대로 연속한 1의 개수는 3이 된다.

16진수 01은 00000001로 표시되고, 최대로 연속한 1의 개수는 1이 된다.

16진수 00은 00000000으로 표시되고, 최대로 연속한 1의 개수는 0이다.

[제약사항]

- 16진수의 알파벳 표시부분은 영문자 **대문자**로 주어진다.
- 1은 최대 9개 까지 연속할 수 있다. 이외의 입력은 주어지지 않는다.

[입력]

- 첫줄에 전체 테스트케이스의 개수 T가 주어진다.
- 각 테스트케이스의 첫 줄에 16진수의 길이 N이 주어진다.
다음 줄에 길이 N의 16진수 문자열이 주어진다. ($2 \leq N \leq 20$)

[출력]

- 각 테스트 케이스의 번호를 '#'과 함께 출력하고, 빈칸에 이어 답을 출력한다.

[Python 트랙] 6회차 – 알고리즘



[입력 예시]

```
3
3
E6D
2
FF
4
5BBD
```

(algo1_sample_in.txt 참고)

[출력 예시]

```
#1 3
#2 8
#3 4
```

(algo1_sample_out.txt 참고)



[Python 트랙] 6회차 – 알고리즘

| 문제2 : 싸피 로봇 (배점 : 30점)

싸피에서도 로봇을 이용해 각 반에 필요한 물품을 전달하기로 했다. 로봇은 사무국을 출발해 각 강의실을 돌고 다시 사무국으로 돌아와야 한다. 사무국을 출발해 각 강의실을 한 번씩만 방문하고 사무실로 돌아올 때의 최소 배터리 사용량을 알아내 배터리 용량을 결정하려고 한다. **이 때 두 강의실 a, b는 반드시 a를 먼저 들른 후 b를 방문해야 한다고 한다. a와 b사이에 또다른 강의실을 거치는 것은 가능하다.**

0번은 사무국, 1번부터 N-1번은 강의실 번호이다. 사무국과 강의실 사이, 강의실과 강의실 사이는 이동 방향에 따라 배터리 소비량이 다를 수 있다.

i에서 j로 이동하는 소비량 e_{ij} 는 $N \times N$ 행렬 형태로 주어진다.

다음은 $N=3$ 인 경우 각 구간의 배터리 소비량이다.

e	0	1	2	j
0	0	18	34	
1	48	0	55	
2	18	7	0	
i				

$a=2, b=1$ 인 경우 가능한 경로는 0-2-1-0 뿐이며, 필요한 배터리 용량은 $e[0][2] + e[2][1] + e[1][0] = 89$ 가 된다



[Python 트랙] 6회차 – 알고리즘

[입력]

첫 줄에 테스트케이스의 수 T가 주어진다. ($1 \leq T \leq 10$)

다음 줄부터 각 케이스 별로 첫 줄에 N, N개씩 N개의 줄에 걸쳐 e_{ij} 가 주어지고, 마지막 줄에 a와 b가 주어진다. ($3 \leq N \leq 10$, $1 \leq e_{ij} \leq 100$, $1 \leq a, b < N$, $a \neq b$)

[출력]

과 테스트 케이스 번호를 출력하고, 빈칸에 이어 답을 출력한다.

[입력 예시]

```
3
3
0 18 34
48 0 55
18 7 0
2 1
4
0 83 65 97
82 0 78 6
19 19 0 82
6 34 94 0
1 3
5
0 9 26 85 42
14 0 84 31 27
58 88 0 16 46
83 61 94 0 17
40 71 24 38 0
3 1
(algo2_sample_in.txt 참고)
```

[출력 예시]

```
#1 89
#2 96
#3 144
(algo2_sample_out.txt 참고)
```

[Python 트랙] 6회차 – 알고리즘



| 문제 3 : 복잡도 (배점 : 20점)

(1) Big-O 표기법의 의미에 대해 간단히 설명 하시오.

(2) 함수 $f(n)$ 의 계산량 $T(n)$ 이 다음과 같이 표시된다고 한다.

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

$f(n)$ 의 시간 복잡도를 n 에 대한 big-O 표기법으로 표시하고,
이를 구하는 과정을 수식으로 나타 내시오.