



월말 평가_알고리즘

| Background

- ✓ 배열의 인덱스 연산을 활용해 문제를 해결.
- ✓ 2차원에 대한 탐색기법을 이해하고 문제를 해결.

| Goal

- ✓ 문제를 정확히 이해하고 문제를 해결할 수 있다.
- ✓ 탐색 기법을 활용해 문제를 해결한다.

| 환경 설정

- 1) Pycharm (Python 3.6) : 코드 디버깅 툴을 사용해 테스트케이스에 대한 결과를 확인한다.
- 2) 프로젝트 생성 : 적당한 이름(Algo)으로 프로젝트를 생성한다.
- 3) 소스코드 작성 : 아래 2개의 이름으로 파이썬 코드를 작성한다.
Algo1_도시_반_이름.py , Algo2_도시_반_이름.py,
(도시: 서울, 구미, 대전, 광주)
- 4) 서술형 문제 : Algo3_도시_반_이름.txt 로 작성한다.
- 5) 제출 파일 : 위 3개의 작성한 파일만 Algo 도시 반 이름.zip 으로 압축하여 제출한다.
(탐색기에서 파일 선택 후 오른쪽 클릭 - 보내기 - 압축(zip)폴더 선택)
(edu.ssafy.com 사이트에 업로드)
- 6) 주석 : 제출 소스 코드에 적절한 주석이 기술되어야 한다. 주석도 점수에 반영되며 주석이 없으면 해당 문제는 0점 처리 될 수 있다.
- 7) 테스트 케이스 : 모든 테스트 케이스는 공개되지 않으며, 부분적으로 제공됩니다.
- 8) 채점 : 테스트 케이스별로 부분 채점 된다.
- 9) 배점 : 각 문제의 배점이 다르므로 표기된 배점을 반듯이 확인한다.
1번 문제(60점), 2번 문제(30점), 3번 문제(10점)

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



월말 평가_알고리즘

[문제1] 부분 사각형 회전 (배점 : 60점)

크기가 $N \times N$ 인 사각 영역 내에서 시작 위치가 (X, Y) 이고 가로세로가 K 인 부분 사각형을 C 도 만큼 회전시키려고 한다. 회전 후 R 번째 행의 합을 구하여 출력 하시오. 회전 각도는 90도 간격(90/180/270/360/450~)으로 90부터 1800까지의 수로 입력된다.

예를 들어 크기가 5×5 인 사각형에서 K 가 3이고 X, Y 는 각 2와 3인 경우 90도, 180도 회전하면 다음과 같다.

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	22	17	12	15
4	16	23	18	13	20
5	21	24	19	14	25

90도 회전 => 77

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	24	23	22	15
4	16	19	18	17	20
5	21	14	13	12	25

180도 회전 => 95

회전 후 3번째 행의 합은, 90도 회전 시에는 $11+22+17+12+15=77$, 180도 회전 시에는 $11+24+23+22+15=95$ 이다.

단, 부분 사각형이 시작 위치를 기준으로 $N \times N$ 영역을 벗어나는 경우에는 -1을 출력한다. 다음과 같이 X, Y 가 (2, 4)이고 K 가 3이면 영역을 벗어나므로 -1을 출력한다.

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25

=> -1



월말 평가_알고리즘

[입력]

첫 줄에 테스트케이스 개수 T 가 주어진다. ($1 \leq T \leq 10$)

다음 줄부터 테스트 케이스의 첫 줄엔 배열의 크기 N , 회전 각도 C , 시작 위치 X , Y , 부분 영역의 가로세로 값 K , 출력 행 R 이 주어진다.

($2 \leq N \leq 10$, $1 \leq X, Y, K, R \leq N$, $0 < C \leq 1800$)

그 다음 줄부터 N 줄에 걸쳐 각행 별로 N 개의 자연수가 공백으로 구분되어 주어진다. 주어지는 자연수는 100이하이다.

[출력]

각 줄에 #과 1부터인 테스트케이스번호, 빈칸 후에 R 행의 합을 출력하시오.

[입력 예시]

```
3
5 90 2 3 3 3
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
5 630 2 3 3 3
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
5 1800 2 4 3 3
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

[출력 예시]

```
#1 77
#2 83
#3 -1
```



월말 평가_알고리즘

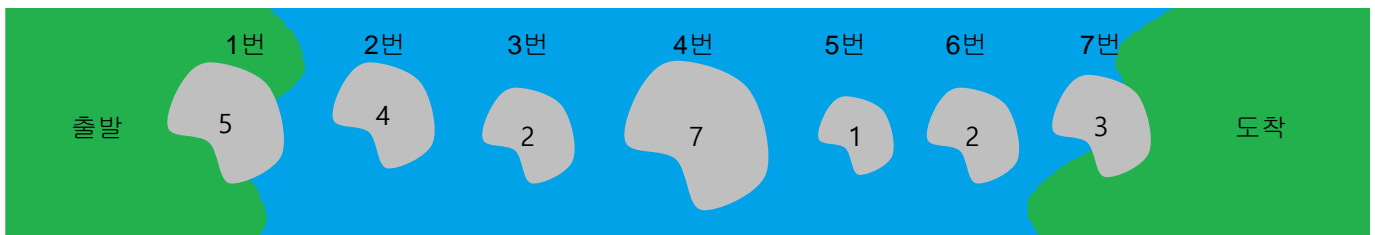
[문제2] 징검다리 (배점 : 30점)

시골에 사는 길동이의 집으로 가는 길에는 돌의 개수가 N개인 징검다리가 있다. 평소 길동이 징검다리의 돌의 크기를 기준으로 해서 점수를 매겨 두었다. 길동이는 재미있게 징검다리를 건너기로 했다. 길동이는 M번 만큼만 돌을 밟고 징검다리를 건너려고 한다. 이 때, 얻을 수 있는 최대 점수는 몇 점인가?

징검다리를 건널 때는 다음과 같은 규칙을 따라야 한다.

- 돌을 밟으면 점수를 얻는다.
- 징검다리를 건널 때는 돌을 밟거나 건너 뛸 수 있다.
- 연속된 두 개의 돌을 뛰어 넘을 수 없다.
- 주어진 횟수 M번 만큼 돌을 밟아야 한다.
- 뒤로 되돌아 갈 수는 없다.

[예시] N = 7, M = 5 일 때,



1번, 2번, 4번, 6번, 7번 순서대로 돌을 밟으면 총 21점으로 최대점이다.

[입력]

첫 줄에는 테스트케이스의 개수가 주어진다.

둘째 줄부터는 테스트케이스가 주어진다. 테스트케이스의 첫 줄에는 $N(1 \leq N \leq 20)$,

$M(N/2 \leq M \leq N)$ 이 공백으로 구분되어 주어진다. 테스트 케이스의 둘째 줄에는 징검다리의 돌의 크기가 공백으로 구분되어 N개 만큼 주어진다. 돌의 크기는 1이상 10이하 이다.

[출력]

각 줄에 #과 1부터 시작하는 테스트 케이스의 번호와 최대점수를 띄어쓰기로 구분하여 한 줄에 출력하라.

월말 평가_알고리즘



[입력 예시]

```
3
7 5
5 4 2 7 1 2 3
8 6
10 10 1 3 1 5 4 7
3 1
1 3 1
```

[출력 예시]

```
#1 21
#2 39
#3 3
```

월말 평가_알고리즘



[문제3] 다음에 제시하는 이진검색 알고리즘에 대한 호출 횟수 $T(n)$ 을 입력값 n 에 대해 표현하고 초기 표현식에서 시간복잡도(O)를 구하기 위한 풀이 과정을 기술하시오. (배점 : 10점)

(거듭제곱은 \wedge 를 사용해서 n^2 형태로 표현 하시오.)

[제시] python 이진탐색 알고리즘

```
def binarySearch2(a, low, high, key) :  
    if low > high : # 검색 실패  
        return False  
    else :  
        middle = (low + high) // 2  
        if key == a[middle] : # 검색 성공  
            return True  
        elif key < a[middle] :  
            return binarySearch2(a, low, middle-1, key)  
        elif a[middle] < key :  
            return binarySearch2(a, middle+1, high, key)
```

(참고) 순차 검색 알고리즘의 정답 예.

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

$$= T(n-2) + 1 + 1$$

$$= T(1) + 1 + \dots + 1$$

$$T(n) = O(n)$$