



(A반)과목평가_알고리즘 기본

| 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_지역_반_이름.py , **Algo2_지역_반_이름.py**,
(지역: 서울, 구미, 대전, 광주)
- 4) 서술형 문제 : **Algo3_지역_반_이름.txt** 로 작성한다.
- 5) 제출 파일 : 위 3개의 작성한 파일만 **Algo 지역 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 제공된다.
- 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 테스트 볼 것 (부정 행위시 강력 조치 및 근거가 남음)

코드가 유사하여 부정 행위로 판단 시 모두 **0점** 처리 됨



(A반)과목평가_알고리즘 기본

[문제1] 사각 도너츠의 합(배점 : 60점)

[제출파일명] Algo1_지역_반_이름.py

싸린이는 일타싸피 상금으로 싸피도너츠 쿠폰을 받았다. 싸린이는 사각모양의 도너츠를 좋아한다. 그래서 도너츠 모양의 문제를 풀어 보려고 한다.

크기가 $N \times M$ 인 배열에 자연수가 적혀 있다. 한 변의 길이가 K 인 사각 테두리의 숫자 합계 중 최대값과 최소값의 차를 구하려고 한다.

예를 들어 크기가 3×3 인 배열에 아래그림과 같이 숫자가 주어지고 한 변의 길이 K 가 3인 경우, 하나의 사각 테두리만 존재하고, 합은 $1+2+3+6+9+8+7+4 = 40$ 이다.

최대값 - 최소값은 0이 된다.

	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

또 다른 예를 들어 보면, 크기가 4×4 인 배열에 K 가 3인 경우는 총 4가지의 경우가 있을 수 있다.

	1	2	3	4
1	2	3	4	3
2	5	6	7	8
3	9	7	9	7
4	1	2	4	5

	1	2	3	4
1	2	3	4	3
2	5	6	7	8
3	9	7	9	7
4	1	2	4	5

	1	2	3	4
1	2	3	4	3
2	5	6	7	8
3	9	7	9	7
4	1	2	4	5

	1	2	3	4
1	2	3	4	3
2	5	6	7	8
3	9	7	9	7
4	1	2	4	5

사각 테두리의 합은 각각 46, 47, 43, 46이다.

$N \times M$ 배열과 K 값이 주어질 때, 사각 테두리 합의 최대값과 최소값의 차를 구하라. 위의 예제는 $47 - 43 = 4$ 가 된다.



(A반)과목평가_알고리즘 기본

[입력]

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

다음 줄부터 테스트 케이스의 첫 줄엔 배열의 행의 개수 N 과 배열의 열의 개수 M , 사각 테두리 한변의 길이 K 가 주어진다. ($3 \leq K \leq N, M \leq 20$)

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

[출력]

각 줄에 #과 1부터인 테스트케이스번호, 빈칸후에 사각테두리의 숫자의 합 중 최대값과 최소값의 차를 출력하시오.

[입력 예시]

```
3
3 3 3
1 2 3
4 5 6
7 8 9
4 4 3
2 3 4 3
5 6 7 8
9 7 9 7
1 2 4 5
6 5 4
11 75 97 9 36
14 33 72 12 57
44 77 38 98 67
38 30 69 16 48
45 29 35 64 56
23 75 48 87 45
```

[출력 예시]

```
#1 0
#2 4
#3 198
```



(A반)과목평가_알고리즘 기본

[문제2] 연쇄 폭발(배점 : 30점)

[제출파일명] Algo2_지역_반_이름.py

싸피국은 파이썬국과 전쟁을 하고 있다. 서로 교전중인 전장에는 많은 폭탄들이 매설되어 있다. 싸피국은 최근 전장에 매설된 폭탄들의 정보가 기록된 지도를 획득했다. 지도는 $N \times N$ 영역으로 0은 폭탄이 없는 곳이고, 0보다 큰 값은 폭탄이 매설된 곳이다.

폭탄에 해당하는 값은 폭탄의 폭발력을 의미하고, 폭탄이 터지는 경우 폭발력이 미치는 범위를 의미한다.

아래 그림 1은 4×4 영역에 대한 예제이다.

0	0	1	0
0	2	0	1
0	0	0	0
1	1	0	2

<그림1>

		2	

<그림2>

폭탄의 폭발력이 2인 경우에는 그림 2와 같이 상하좌우 방향으로 2칸의 범위에 폭발력이 미친다. 폭발의 범위가 지도의 경계를 벗어나면 폭발력은 사라진다.

폭탄이 폭발했을 때 폭발 범위에 다른 폭탄이 있다면 연쇄적으로 폭발하게 된다.

그림 3에서 (3행, 3열)에 위치한 폭탄이(노란색) 폭발하는 경우를 생각해 보자.

(3행, 3열) 폭탄이 폭발하면 폭발 범위 내에 위치한 (1행, 3열), (3행, 1열)의 폭탄(연두색)이 연쇄적으로 폭발하게 된다(그림 4).

그리고, 다음으로 (3행, 1열)의 폭탄에 의해 (3행, 0열)의 폭탄(회색)도 폭발한다.(그림 5)

0	0	1	0
0	2	0	1
0	0	0	0
1	1	0	2

<그림 3>

0	0	1	0
0	2	0	1
0	0	0	0
1	1	0	2

<그림 4>

0	0	1	0
0	2	0	1
0	0	0	0
1	1	0	2

<그림 5>

따라서, 모두 4개의 폭탄이 폭발하게 된다. $N \times N$ 영역에 대한 폭탄들의 정보와 첫 번째로 폭발하는 폭탄의 위치가 주어질 때, 연쇄적으로 폭발하게 되는 폭탄의 총수를 계산해보자.

(A반)과목평가_알고리즘 기본



[입력]

첫 줄에 테스트 케이스 개수 T 가 주어진다. $1 \leq T \leq 10$

다음 줄부터 테스트 케이스마다 첫 줄에는 지도의 크기 N 이 주어진다. ($4 \leq N \leq 20$)

다음 줄에 처음 폭발하는 폭탄의 행과 열의 위치 R, C 가 주어진다. ($0 \leq R, C < N$)

다음 N 개의 줄마다 공백으로 구분된 N 개의 정수들이 주어진다. ($1 \leq \text{폭발력} \leq 10$)

[출력]

#과 1번부터인 테스트케이스 번호를 출력하고, 폭발하는 폭탄의 총 개수를 출력한다.

[입력 예시]

```
3
4
3 3
0 0 1 0
0 2 0 1
0 0 0 0
1 1 0 2
6
3 2
0 0 0 0 0 0
0 0 3 0 1 1
1 0 0 0 0 0
0 0 3 0 0 2
2 0 0 0 0 0
0 1 0 2 0 2
10
8 7
0 3 0 0 3 0 0 0 0 0
0 3 0 0 0 0 0 1 0 3
0 0 5 0 0 0 0 0 3 0
0 0 0 0 0 0 0 0 0 0
0 5 0 0 0 2 0 5 0 0
0 0 0 0 0 3 0 2 0 4
4 0 2 0 0 2 1 4 0 0
0 0 0 0 0 5 0 0 0 0
0 0 0 0 0 0 4 5 5 1
3 0 3 0 2 4 0 0 0 1
```

[출력 예시]

```
#1 4
#2 8
#3 22
```

(A반)과목평가_알고리즘 기본



[문제3] 서술형 (배점 : 10점)

[제출파일명] Algo3_지역_반_이름.txt

길동이는 최소힙을 이용하여 프로그래밍을 하려고 한다. 최소힙은 트리내 모든 노드에 대해 부모 노드의 값이 자식노드의 값 보다 작은 완전이진트리를 의미한다.

길동이는 최소힙을 크기 8인 배열(리스트)로 구현하였고, 정수로 된 데이터를 저장한다.

길동이가 다음과 같은 순서로 최소힙에 데이터를 삽입(push) 하거나 삭제(pop) 하는 작업을 수행 한다고 하였을 때, 배열의 요소에 값을 할당, 삭제, 수정(교환) 하는 과정을 나열 하시오.

[작업순서]

2 삽입, 3 삽입, 4 삽입, 5 삽입, 1 삽입, 9 삽입, 8 삽입, 삭제, 7삽입, 삭제, 1삽입

[제약조건]

- Heap의 최대 노드 개수는 7개이다.
- 배열의 비어 있는 요소는 '-' 으로 표시한다.
- 루트 노드는 배열의 1번 인덱스로 지정한다
- 부모 노드의 인덱스가 M 일 때, 왼쪽 자식 노드, 오른쪽 자식 노드의 인덱스는 각각 $M * 2$, $M * 2 + 1$ 이다.
- 배열의 중간 변화과정은 배열에 데이터가 **삽입/삭제/수정 되는 경우 모두** 나열한다. (힙에서 데이터가 삽입, 삭제 되는 경우 규칙에 맞게 데이터의 자리를 변경하는 과정이 필요하다. 이 경우에 발생하는 모든 삽입/삭제/교환에 대해서 배열 변경과정을 나열한다.)
- 삭제의 경우 삭제된 데이터를 별도로 표시한다.

예시)

초기상태 -, -, -, -, -, -, -

2 삽입 -, 2, -, -, -, -, -

삭제 -, -, -, -, -, - (삭제된 값 2)