

[Python트랙] 과목평가2 - 알고리즘 기본



| Background

- ✓ 배열에 대한 이해와 활용
- ✓ 정렬에 대한 이해와 활용

| Goal

- ✓ 반복문과 조건문을 이용하여 배열의 요소에 접근할 수 있다.
- ✓ 배열에서 특정 조건을 만족하는 연속한 원소를 탐색할 수 있다.
- ✓ 배열을 저장된 자료를 특정 정렬 알고리즘으로 정렬할 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


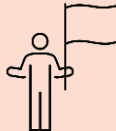



[Python트랙] 과목평가2 - 알고리즘 기본















| 문제1 : 싸피 운동회 (배점 : 40점)

싸피 운동회에는 깃발 게임이 진행된다. 게임은 팀원들이 한 줄로 서서 지시에 따라 정확하게 기를 올리거나 내리는 동작을 하고, 모든 지시가 끝난 후 정확한 상태가 되었는지를 겨룬다. 모두 T개의 팀이 차례로 참가하며, 게임 규칙은 다음과 같다.

- ① 각 팀은 N명의 팀원으로 구성되며, 1번부터 N번까지 번호가 붙은 자리에 한 줄로 늘어선다.
- ② 각 팀원은 지시사항에 따라 기를 들거나 내린 상태로 대기한다.

1	2	3	4	5
				

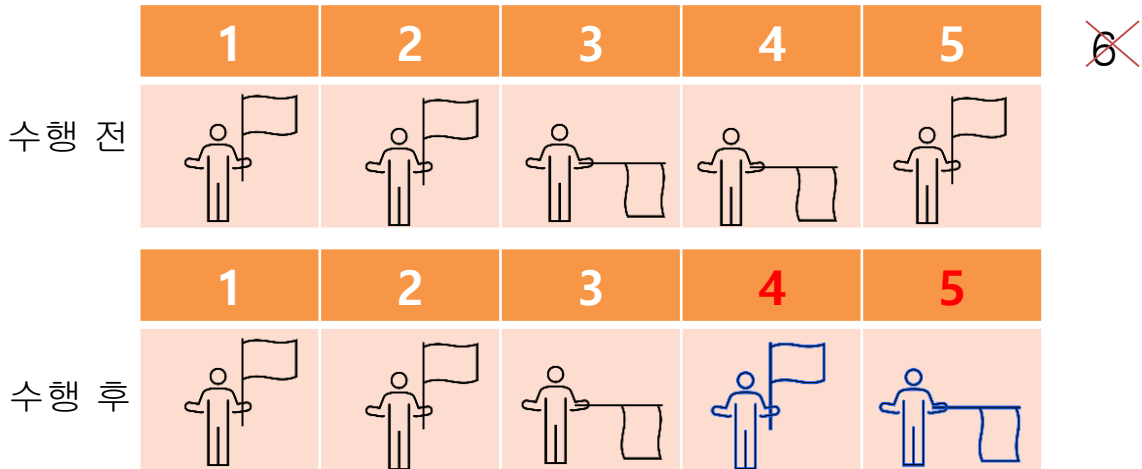
- ③ 준비가 끝나면 진행자로부터 M번의 명령이 전달된다. 각 명령은 정수 a b c로 구성되며, a는 난이도, b는 기준 번호, c는 사람수를 의미하며, 오늘은 예선전이라 난이도 1만 등장한다. 예를 들어 1 3 2는 난이도 1로 3번부터 2명이 기의 상태를 바꾸라는 명령이다. 1 3 2를 수행한 결과는 다음과 같다.

	1	2	3	4	5
수행 전					
					
	1	2	3	4	5
수행 후					

[Python트랙] 과목평가2 - 알고리즘 기본



- ④ 만약 1 4 3이 전달되면, 기준 4번 부터 상태를 바꾸며, 세번째인 6번이 없으므로 4, 5 만 상태를 바꾼다.



[입력]

첫 줄에 팀 수 T, 다음 줄부터 팀 별로 첫 줄에 N과 M, 다음 줄에 빈칸으로 구분된 N명의 초기 상태가 1 또는 0으로 주어진다. 이후 M 줄에 걸쳐 a b c가 주어진다. ($3 \leq T \leq 10$, $3 \leq N \leq 20$, $1 \leq M \leq 10$, $a=1$, $1 \leq b$, $c \leq N$)

[출력]

각 팀 별로 한 줄에 #과 1번부터인 팀 번호, 빈칸에 이어 팀별로 명령을 수행한 결과를 출력한다.

(입출력 예시는 다음 페이지)

[Python트랙] 과목평가2 - 알고리즘 기본



[입력 예시]

```
3
5 1
1 1 0 0 1
1 3 2
5 3
1 1 0 0 1
1 3 2
1 4 1
1 2 4
5 1
1 1 0 1 0
1 4 1
```

(algo1_sample_in.txt 참고)

[출력 예시]

```
#1 1 1 1 1 1
#2 1 0 0 1 0
#3 1 1 0 0 0
```

(alg1_sample_out.txt 참고)

[Python트랙] 과목평가2 - 알고리즘 기본



| 문제2 : RGB 괴물 (배점 : 35점)

10x10 격자 형태로 구분된 지역에 RGB 괴물이 살고 있다. 괴물은 자기 구역을 벗어나지 않지만 광선을 발사하며, 괴물이 있는 구역이나 괴물 광선이 도달하는 구역에 들어가는 사람은 그동안 모은 알고리즘을 잃게 된다고 한다.

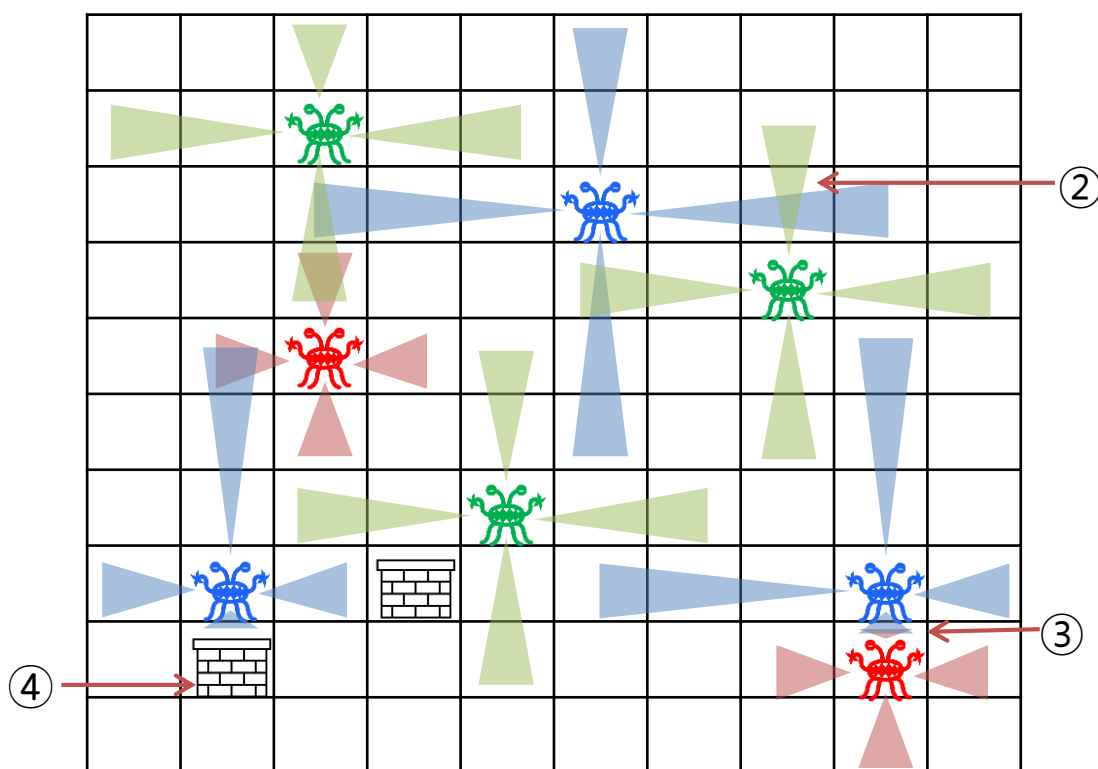


그림1. RGB 괴물이 있는 10x10 구역의 예

괴물 광선은 괴물의 상하좌우 방향으로 발사되며, R괴물은 한 칸, G괴물은 두 칸, B괴물은 3칸을 비춘다.

- ① 한 구역에는 최대 하나의 괴물만 있다.
- ② 광선은 서로 교차한다. (그림1. ② 참고)
- ③ 광선이 비추는 곳에 다른 괴물이 있으면 광선은 거기서 멈춘다. (그림1. ③ 참고)
- ④ 광선은 벽을 통과하지 못한다. (그림1. ④ 참고)

[Python트랙] 과목평가2 - 알고리즘 기본



10x10 구역에 대한 정보가 주어졌을 때, 괴물과 괴물 광선으로 부터 안전한 구역의 개수를 구해보자. 단, 벽이 있는 구역은 사람도 들어갈 수 없어서 안전한 구역에서 제외한다.

[입력]

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

각 테스트케이스 별로, 첫 줄에는 RGB 괴물의 수 N이 주어진다.

($1 \leq N \leq 100$)

다음 10개의 줄에 10x10 영역 정보가 주어진다. 0은 괴물이 없는 구역, 1은 R괴물, 2는 G괴물, 3은 B괴물, 4는 벽을 나타낸다.

[출력]

각 줄에 테스트케이스 별로, #에 이어 테스트케이스 번호, 빈칸에 이어 안전한 구역의 수를 출력한다.

(입출력 예시는 다음 페이지)

[Python트랙] 과목평가2 - 알고리즘 기본



[입력 예시]

```
2
3
0000000000
0000001000
0000000000
0002000000
0000000000
0000000000
0000003000
0000000000
0000000000
0000000000
8
0000000000
0020000000
0000030000
0000000200
0010000000
0000000000
0000200000
0304000030
0400000010
0000000000
```

(algo2_sample_in.txt 참고)

[출력 예시]

```
#1 73
#2 42
```

(algo2_sample_out.txt 참고)

[Python트랙] 과목평가2 - 알고리즘 기본



| 문제3 : (배점 : 25점)

다음은 어떤 정렬에 대한 설명이다.

- 인접한 두 개의 원소를 비교하며 자리를 계속 교환하는 방식
- 내림차순에 대한 의사 코드는 다음과 같다.

```

sort(a, N)::
for i : N-1 -> 1
    for j : 0 -> i-1
        if a[j] < a[j+1]
            a[j] <-> a[j+1]

```

배열 a, 배열의 크기 N
정렬할 구간의 끝
비교할 원소 중 왼쪽 원소의 인덱스
왼쪽 원소가 더 작으면
오른쪽 원소와 교환

3-1. 다음 배열에 대해, 위 코드에서 $i=N-1$ 일 때 ()안에 들어갈 배열 a의 변화를 모두 적으시오. 답은 j값에 따라 줄을 바꿔 표시한다. (N=5)

a	2	7	1	4	3
---	---	---	---	---	---

답안 예)

2 7 1 4 3

() (j==0)

() (j==1)

() (j==2)

...

() (j==i-1)

3.2 이 정렬의 이름과 장단점에 대해 간단히 설명하시오.