

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용



| Background

- ✓ 프로그래밍의 논리적인 이해
- ✓ 다양한 알고리즘 기법과 적용에 대한 이해

| Goal

- ✓ 문제를 분석하고 알맞은 알고리즘 기법을 적용하여 프로그래밍할 수 있다.

| 환경 설정

1) 워크스페이스 설정

Spring Tool Suite 4 시작 시 워크스페이스는 **c:\WAlgoTest** 로 변경한다.
(만약, Spring Tool Suite 4 시작 시 워크스페이스를 물어보는 창이 보이지 않고 바로 실행될 경우 File> Switch Workspace > Other 메뉴를 이용하여 워크스페이스를 변경한다)

2) 프로젝트 생성

적당한 이름으로 프로젝트를 생성한다. (ex: Algo)

3) 답안 작성

- 문제에서 원하는 출력 결과가 나오도록 작성
(단, src 폴더 아래 패키지 없이 작성 : default package)
- 제출파일명 : Test문제번호_지역_반_이름.java
ex) Test1_대전_04반_김싸피.java

4) 파일 제출

완성 후 **파일**만 모아서 “Algo_지역_반_이름.zip”으로 압축하여 업로드한다.
(ex: Algo_대전_04_김싸피.zip)

※ 코드에 주석을 작성한다. (작성하지 않을 시 감점할 수 있음)

성실과 신뢰로 평가에 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

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용

| 문제1 : 다투는 고양이 (배점 30점)

$N \times N$ 크기의 2차원 배열 형태의 들판에 고양이 M 마리가 살고 있습니다. 이 고양이들은 영역 다툼을 통해 자신의 영역을 차지합니다.



[영역 다툼을 하는 고양이들]

각 고양이에게 1부터 M 까지 번호를 붙여, 각 고양이가 차지한 영역을 번호로 표시하면 들판의 초기 상태가 아래와 같이 주어질 수 있습니다.

(한 고양이가 차지하는 영역은 상하좌우로 인접하도록 주어집니다.)

0으로 표시한 영역은 어떤 고양이도 차지하지 않은 중립 영역을 나타냅니다.

1	1	1	1	2
0	0	1	1	3
0	3	3	3	3
4	5	0	0	6
7	7	7	7	7

[고양이가 차지한 영역]

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용

| 문제1 : 다투는 고양이 (배점 30점)

이 고양이들 중 대장 고양이를 찾으려고 합니다. 차지하고 있는 영역이 가장 넓은 고양이가 대장 고양이이며, 영역의 넓이가 같다면 번호가 작은 고양이가 대장 고양이가 됩니다.

1	1	1	1	2
0	0	1	1	3
0	3	3	3	3
4	5	0	0	6
7	7	7	7	7

고양이들이 차지한 영역을 세어 보면 다음과 같습니다.

- 1번 고양이 : 6 칸
- 2번 고양이 : 1 칸
- 3번 고양이 : 5 칸
- 4번 고양이 : 1 칸
- 5번 고양이 : 1 칸
- 6번 고양이 : 1 칸
- 7번 고양이 : 5 칸

따라서 가장 넓은 영역을 차지하고 있는 1번 고양이가 대장 고양이임을 알 수 있습니다.

이처럼 들판의 형태가 주어졌을 때, 대장 고양이를 구하는 프로그램을 작성하세요.

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용



| 문제1 : 다투는 고양이 (배점 30점)

[입력]

가장 첫 줄에는 테스트 케이스의 개수 T 가 주어지고, 그 아래로 각 테스트 케이스가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 들판의 크기 N 과 고양이의 수 M 이 주어지며 다음 N 개의 줄에 걸쳐 N 개의 정수가 주어진다. 0은 중립 영역을 의미하며, 1 이상의 정수는 해당 칸을 차지한 고양이의 번호를 의미한다.

[제약 조건]

- $1 \leq N \leq 20$
- $1 \leq M \leq 10$

[출력]

각 줄마다 "# T "를 출력한 뒤 대장고양이의 번호를 출력한다.
(T 는 테스트 케이스의 번호를 의미하며 1부터 시작한다.)

[입력 예시]

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

[출력 예시]

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





[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용

| 문제2 : 가출한 고양이 (배점 30점 + 10점)

이 문제는 제약조건의 범위에 따라 배점이 달라지는 문제입니다.

김싸피네 고양이는 창문 밖의 나비를 쫓아가다 집에서 너무 멀리 떨어지게 되었습니다. 똑똑한 고양이는 집의 위치를 기억하고 있기 때문에 **최단 경로**로 돌아가려고 합니다. **최단 경로란**, 이동하는 칸 수가 가장 적은 경로를 의미합니다.

하지만 돌아가는 길은 고양이에게 위험하기 때문에 **길고양이 쉼터**를 적어도 한 곳 이상 들리면서 **위험성의 합이 가장 적은** 길을 선택하고자 합니다.

 0	3	7	4	3
2	8	3	 5	8
9	 5	1	4	2
6	2	4	9	6
4	5	3	1	 0

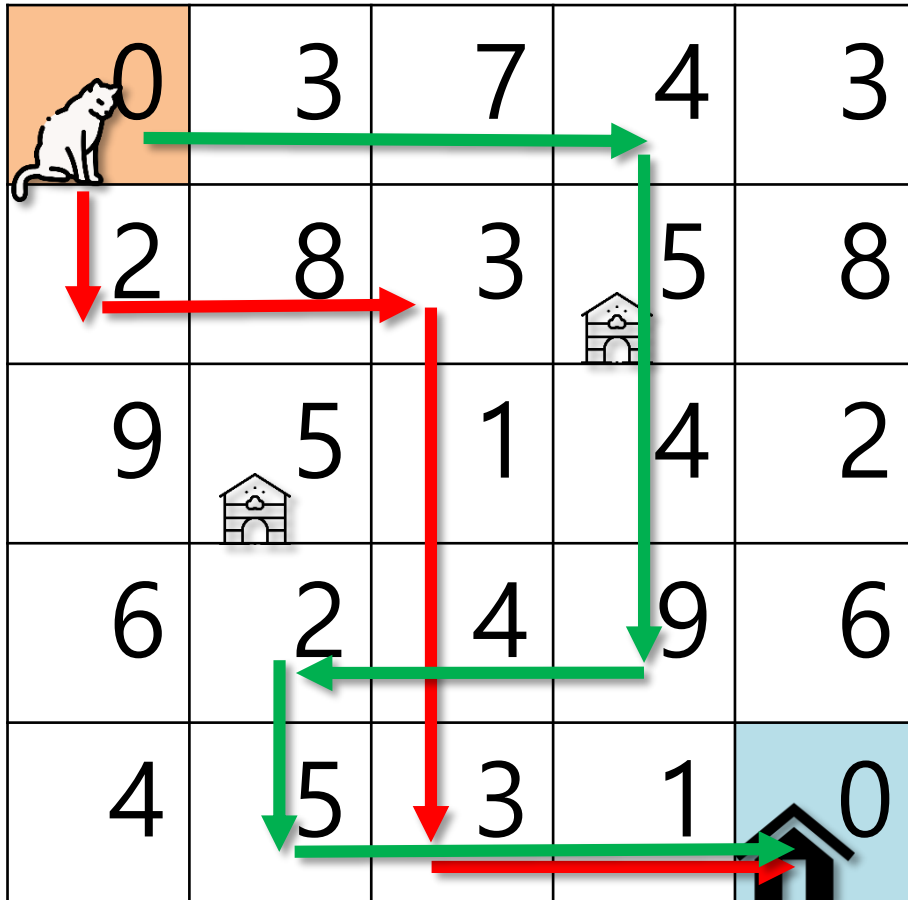
위의 그림은 거리의 위험성을 5 x 5 크기의 2차원 배열 형태로 나타내고 있습니다. (1, 3)와 (2, 1)에는 길고양이 쉼터가 위치하고 있습니다.

고양이는 가장 왼쪽 위(0, 0)에서 출발하여 길고양이 쉼터를 한 곳 이상 거쳐 가장 오른쪽 아래(4, 4)에 있는 집까지 **최단거리**로 이동해야 합니다.

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용

| 문제2 : 가출한 고양이 (배점 30점 + 10점)

아래 그림은 고양이가 집까지 이동하는 여러 가지 방법 중 두 가지를 화살표로 표시하고 있습니다.



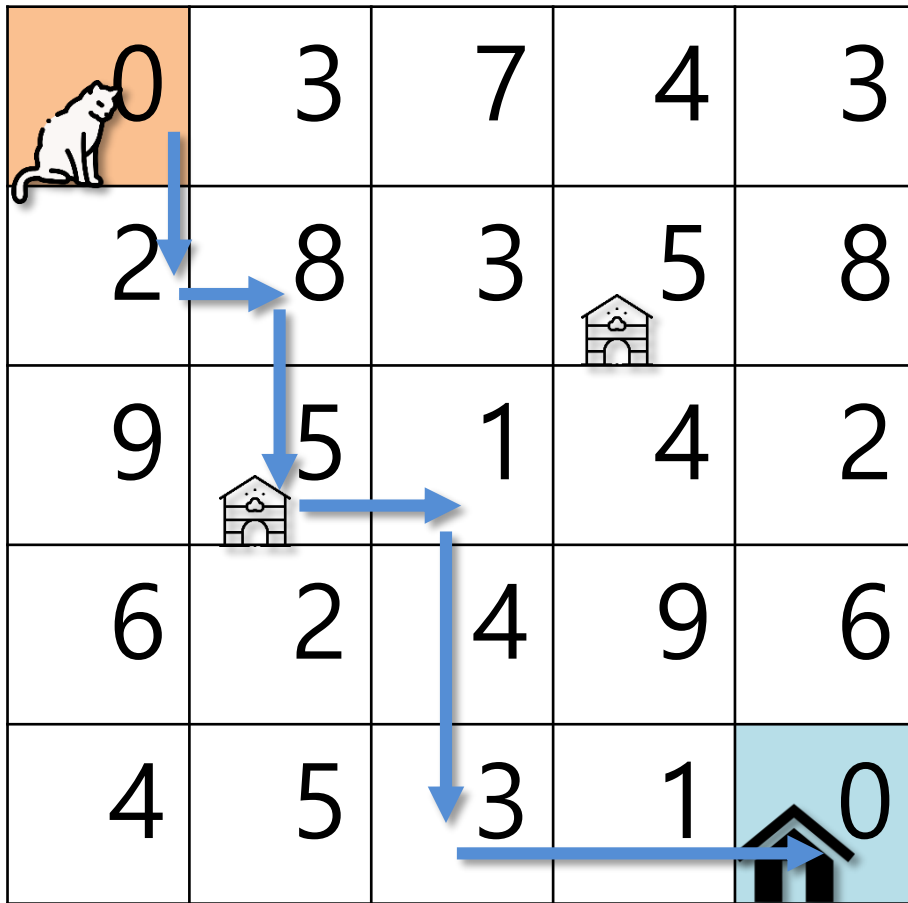
초록색으로 표시한 경로의 경우 최단 경로가 아니므로 이동할 수 없는 경로입니다.

빨간색으로 표시한 경로로 이동하게 되면 위험성의 합은 $0 + 2 + 8 + 3 + 1 + 4 + 3 + 1 + 0 = 22$ 이며, 이보다 더 위험성이 적은 경로로 이동할 수는 없습니다. 하지만 길고양이 쉼터를 거치지 않았기 때문에 이동할 수 없는 경로입니다.

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용

| 문제2 : 가출한 고양이 (배점 30점 + 10점)

파란색으로 표시한 경로의 경우 위험성의 합은 $0 + 2 + 8 + 5 + 1 + 4 + 3 + 1 + 0 = 24$ 이며 (2, 1)에 위치한 길고양이 쉼터를 지나고 있습니다. 길고양이 쉼터를 지나면서 이보다 더 위험성이 적은 경로로 이동할 수는 없습니다.



이처럼, 거리의 위험성이 $N \times N$ 크기의 2차원으로 배열로 주어지고 길고양이 쉼터의 좌표가 주어질 때, 고양이가 길고양이 쉼터를 지나면서 위험성이 가장 적은 경로로 이동할 때의 위험성을 출력하는 프로그램을 작성하세요.

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용



| 문제2 : 가출한 고양이 (배점 30점 + 10점)

[입력]

가장 첫 줄에는 테스트 케이스의 개수 T 가 주어지고, 그 아래로 각 테스트 케이스가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 마을의 크기 N 과 길고양이 쉼터의 개수 M 이 주어진다. 다음 N 개의 줄에 걸쳐 위험성을 나타내는 N 개의 정수가 공백으로 구분되어 주어진다. 그 다음 M 개의 줄에 걸쳐 길고양이 쉼터의 좌표 r 와 c 가 주어진다.

- r 은 행의 번호, c 은 열의 번호를 의미한다.
- 가장 왼쪽 위의 좌표가 $(0, 0)$ 이다.

[제약 조건 1] (배점 30점)

- $1 \leq N, M \leq 10$
- $0 \leq (\text{위험성}) \leq 100$

[제약 조건 2] (배점 +10점)

- $1 \leq N, M \leq 1000$
- $0 \leq (\text{위험성}) \leq 100$

[출력]

각 줄마다 "#T"를 출력한 뒤 고양이가 집에 도착할 수 있는 경로 중 가장 위험성이 적은 경우의 위험성을 출력한다.

(T 는 테스트 케이스의 번호를 의미하며 1부터 시작한다.)

[입력 예시]

```
2
5 2
0 3 7 4 3
2 8 3 5 8
9 5 1 4 2
6 2 4 9 6
4 5 3 1 0
1 3
2 1
3 2
9 9 9
9 0 9
9 9 9
0 2
2 0
```

[출력 예시]

```
#1 24
#2 45
```


| 문제3 : 날리는 고양이 (배점 30점)

고양이 냥싸피는 냥냥펀치를 연속으로 날릴 수 있도록 매일 수련을 하고자 합니다.



[냥냥펀치를 수련 중인 냥싸피]

하지만 첫 날부터 냥냥펀치를 연속해서 여러 번 날리는 것은 어렵기 때문에 매일 횟수를 늘려나가야 합니다.

수련 횟수를 늘려가는 규칙은 다음과 같습니다.

1일째 : 1 회

2일째 : X 회

3일째 : $X + 1$ 회

4일째 : $X + X + 1$ 회

...

K 일째 : $(K - 2\text{일째 연습 횟수}) + (K - 1\text{일째 연습 횟수})$

10일째 되는 날 냥싸피가 연속으로 N 회 이상의 냥냥펀치를 날리기 위해 2일째 최소 몇 번의 냥냥펀치를 날려야 하는지 계산하는 프로그램을 작성하세요.

예를 들어 10일째 되는 날 80 회 이상 냥냥펀치를 날리기 위해 2일째 최소 몇 번의 냥냥펀치를 날려야 하는지 구하는 경우를 생각해보면,

2일째 3회 수련을 하면 10일 동안 순서대로 1, 3, 4, 7, 11, 18, 29, 47, 76, 123 회 냥냥펀치를 날리게 되어 10일째 80 회 이상이 됩니다.

그런데 2일째 2회 수련을 해도 10일 동안 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 회 냥냥펀치를 날리므로 80회 이상이 됩니다.

최소 2회 수련을 해야 10일 째 되는 날 80번 이상의 냥냥펀치를 날릴 수 있게 됩니다.

[자바비전공 트랙] 7회차 과목평가 – 알고리즘 응용



| 문제3 : 날리는 고양이 (배점 30점)

[입력]

가장 첫 줄에는 테스트 케이스의 개수 T 가 주어지고, 그 아래로 각 테스트 케이스가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 10일 째 되는 날 목표로 하는 냥냥펀치 횟수 N 이 주어집니다.

[제약 조건]

- $55 \leq N \leq 1,000,000,000,000$

[효율성 조건]

채점용 테스트 케이스를 입력했을 때, 1초 이내에 실행되어야 함

[출력]

각 줄마다 "#T "를 출력한 뒤 10일 째 N 회 이상의 냥냥펀치를 날리기 위해서 2일째 수련해야 하는 최소 횟수를 출력한다.

(T 는 테스트 케이스의 번호를 의미하며 1부터 시작한다.)

[입력 예시]

3
80
128
58323

[출력 예시]

#1 2
#2 4
#3 1715