

[마이스터 트랙] 월말평가2 - 알고리즘 기본



| Background

- ✓ 알고리즘 해결기법에 대한 이해와 활용
- ✓ 조합과 순열에 대한 이해와 활용

| Goal

- ✓ 다양한 완전탐색 기법을 이용하여 문제를 해결할 수 있다.
- ✓ 순열과 조합에 대해서 자세하게 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 폴더, 프로젝트, 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 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 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

[마이스터 트랙] 월말평가2 - 알고리즘 기본



| 문제1: 나루터 항해(배점 : 40점)

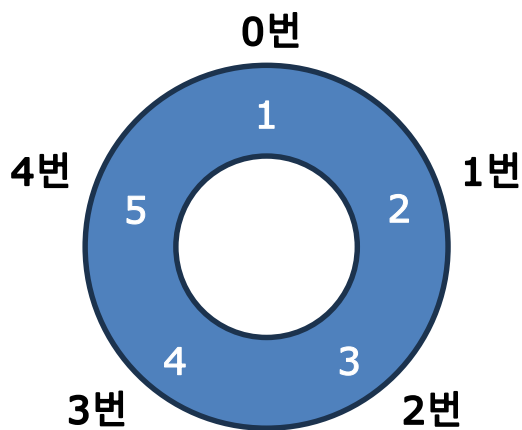
원형 형태의 강을 따라, n개의 나루터가 일정한 간격으로 배치되어 있습니다. 김싸피는 각 나루터마다 일정한 양의 모터를 충전할 수 있습니다. 각 나루터에서 다음 나루터로 이동하기 위해 필요한 에너지는 정해져 있고, 김싸피는 이 강을 한 바퀴 도는 것을 목표로 합니다.

시작하는 시점에서 에너지는 없으며 출발점에서 에너지를 충전한 뒤 인접한 나루터로 출발합니다. 이 때, 해당 나루터에서 요구하는 이동 에너지만큼의 에너지가 충전되어 있지 않으면 더 이상 이동하지 않습니다.

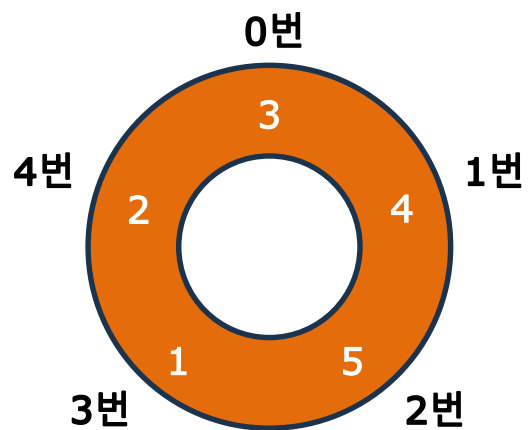
새 나루터에 도착하면 그 나루터에서 제공하는 에너지를 추가로 저장할 수 있습니다.

특정 나루터에서 출발하며, 출발점에 다시 돌아왔을 때 남은 에너지를 출력하세요.

에너지가 부족해 출발점에 다시 돌아올 수 없다면 -1을 반환하세요.



나루터별 충전 에너지량



나루터별 이동 시 필요 에너지

[마이스터 트랙] 월말평가2 - 알고리즘 기본



| 문제1: 나루터 항해(배점 : 40점)

예) 시작 나루터 3번이 주어졌을 때,

- 3번 시작: 에너지 = 0 + 3번 충전 에너지(4) = 4 (남은 에너지)

- 3번 → 4번:

남은 에너지 4는 3번의 필요 에너지(1) 보다 충분함으로 이동 후 3 이 남는다.

이동 후 에너지는 기존(3) + 4번 충전 에너지(5) 으로 8 이 남는다.

- 4번 → 0번:

남은 에너지 8은 4번의 필요 에너지(2) 보다 충분함으로 이동 후 6 이 남는다.

이동 후 에너지는 기존(6) + 0번 충전 에너지(1) 으로 7이 남는다.

- 0번 → 1번:

남은 에너지 7은 0번 필요 에너지(3) 보다 충분함으로 이동 후 4 가 남는다.

이동 후 에너지는 기존(4) + 1번 충전 에너지(2) 으로 6이 남는다.

- 1번 → 2번:

남은 에너지 6은 1번 필요 에너지(4) 보다 충분함으로 이동 후 2 가 남는다.

이동 후 에너지는 기존(2) + 2번 충전 에너지(3) 으로 5이 남는다.

- 2번 → 3번:

남은 에너지 5는 2번 필요 에너지(5) 보다 충분함으로 이동 후 0 가 남는다.

=> 한 바퀴를 돈 후의 남은 에너지인 0을 반환한다.

[마이스터 트랙] 월말평가2 - 알고리즘 기본



[입력]

첫 줄에 테스트케이스 개수 T , 다음 줄부터 테스트 케이스 별로 첫 줄에 나루터 개수 N 과 출발 나루터 M , 다음 줄에 N 개의 충전 에너지(charges) 와 그 다음 줄에 N 개의 이동 시 필요 에너지(costs)가 주어진다.

- $N == \text{charges.length} == \text{costs.length}$
- $1 \leq N \leq 10^5$
- $0 \leq M \leq N$
- $0 \leq \text{gas}[i], \text{cost}[i] \leq 10^4$

[출력]

각 테스트케이스별로 한 줄에 #과 케이스번호, 빈칸에 이어 답을 출력한다.
한 바퀴를 돌 수 있는 출발점이 존재하지 않으면 -1을 출력하세요.

[입력 예시]

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

(algo1_sample_in.txt 참고)

[출력 예시]

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

(algo1_sample_out.txt 참고)

[마이스터 트랙] 월말평가2 - 알고리즘 기본



| 문제2 : 과수원 운영 (배점 : 35점)

김싸피는 대규모 과수원을 운영하고 있습니다. 이 과수원에는 총 N 개의 과일나무가 일렬로 심어져 있으며, 각 나무는 서로 다른 종류의 과일을 맺습니다. 수확철을 맞아 시장에 내다 팔기 위해 최소한 X kg 이상의 과일을 확보해야 합니다. 각 나무를 수확하기 위해서는 일정한 노동력이 필요하고, 수확하면 Y kg 과일을 얻을 수 있습니다.

단, 나무를 수확하지 않으면 그 나무의 과일은 전혀 얻지 못하고, 노동력도 들지 않습니다. 김싸피는 X kg 이상의 과일을 얻기 위해 어떤 나무들을 수확할지 결정해야 합니다.

이 때, 필요한 노동력 비용의 총합을 최소화하여야 합니다. 즉, X kg 이상의 과일을 확보 가능한 모든 수확 방법 중 총 노동력 비용이 가장 작은 값을 찾아야 합니다. 만약 어떤 조합으로도

X kg 이상을 달성할 수 없다면 -1을 출력합니다.



수확 무게	2kg	5kg	3kg	4kg
노동력	3	7	2	6

예) 위와 같이 과일 4개가 주어지고, 목표 수확 무게가 8 Kg인 경우

수박, 바나나만 수확: 과일 합 = $5 + 3 = 8$, 노동력 비용 = $7 + 2 = 9$

귤, 수박, 바나나만 수확: 과일 = $2 + 5 + 3 = 10$, 노동력 비용 = $3 + 7 + 2 = 12$

귤, 바나나, 사과만 수확: 과일 = $2 + 3 + 4 = 9$, 노동력 비용 = $3 + 2 + 6 = 11$

여러 조합 중 X kg 이상을 확보하기 위한 최소 노동력 비용은 9(수박, 바나나) 입니다.

[마이스터 트랙] 월말평가2 - 알고리즘 기본



[입력]

첫 줄에 테스트케이스 개수 T , 다음 줄부터 테스트 케이스별로 첫 줄에 N 과 X , 다음 줄에 N 개의 무게정보(fruits)와 그 다음 줄에 N 개의 노동력(works)가 주어진다.

- $1 \leq N \leq 20, 1 \leq X \leq 200,000$
- $0 \leq \text{fruits}[i], \text{works}[i] \leq 10^4$

[출력]

각 테스트케이스 별로 한 줄에 #과 케이스번호, 빈칸에 이어 답을 출력한다.
목표 과일 X kg 이상을 달성하는 조합 중 최소 노동력을 출력합니다.
달성이 불가능한 경우에는 -1을 출력합니다.

[입력 예시]

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

(algo2_sample_in.txt 참고)

[출력 예시]

```
# 1 9
# 2 11
# 3 -1
```

(algo2_sample_out.txt 참고)

[마이스터 트랙] 월말평가2 - 알고리즘 기본



| 문제3 : 순열과 조합(배점 : 25점)

- 1) 순열과 조합에 대해서 각각 설명하고, 순열과 조합의 차이를 설명하세요.
- 2) 순열과 조합의 적절한 예시를 각각 제시하고, 그 이유를 설명하세요.