

[A반] 4회차 월말평가 – 알고리즘 응용



| Background

- ✓ 탐욕 기법을 이해하고 활용
- ✓ 분할 정복에 대해 이해하고 활용
- ✓ 그래프 알고리즘을 이해하고 활용
- ✓ 완전검색에 대해 이해하고 활용

| Goal

- ✓ 문제를 정확히 이해하고 알고리즘을 설계할 수 있다.
- ✓ 탐욕기법을 적용해 문제를 해결할 수 있다.
- ✓ 그래프 최적화 알고리즘을 적용해 문제를 해결할 수 있다.
- ✓ 완전 검색을 활용해 문제를 해결할 수 있다.

| 환경 설정

1) Pycharm(Python3.5이상)을 이용해서 코드를 작성하고 결과를 확인한다.

- 새로운 Pycharm 프로젝트를 생성 후 코드를 작성한다.

2) 파일 이름 및 제출 방법

- 1,2,3번 문제에 대한 소스 파일은 Algo문제번호_지역_반_이름.py로 만든다.
- 4번은 텍스트 파일로 작성한다.

Algo1_서울_1반_이싸피.py

Algo2_서울_1반_이싸피.py

Algo3_서울_1반_이싸피.py

Algo4_서울_1반_이싸피.txt

- 위 4개의 파일만 지역_반_이름.zip으로 압축하여 제출한다.

서울_1반_이싸피.zip

(탐색기에서 파일 선택 후 오른쪽 클릭 – 보내기 – 압축(zip)폴더 선택)

(edu.ssafy.com 사이트에 업로드)

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 30점, 2번 30점, 3번 30점, 4번 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

[A반] 4회차 월말평가 – 알고리즘 응용



| 문제 1 : 베이비진 게임 (배점 30점)

평소에 카드 놀이를 좋아하는 김싸피는 베이비진이라는 게임을 하려고 한다.

베이비진 게임은 0~9 사이의 숫자 카드에서 임의의 카드 6장을 뽑았을 때, 3장의 카드가 연속적인 번호를 갖는 경우를 run이라 하고, 3장의 카드가 동일한 번호를 갖는 경우를 triplet이라고 한다.

그리고, 6장의 카드가 run과 triplet로만 구성된 경우를 baby-gin이라 한다.

예를 들어,

667767은 두 개의 triplet이므로 baby-gin이다. (666, 777)

054060은 한 개의 run과 한 개의 triplet이므로 역시 baby-gin이다. (456, 000)

101123은 한 개의 triplet가 존재하나, 023이 run이 아니므로 baby-gin 이 아니다. (123을 run으로 사용하더라도 011이 run이나 triplet가 아님)

6자리의 숫자를 입력 받아 baby-gin이면 1을 출력하고 아니면 0을 출력하시오.

[입력]

첫 줄에 테스트케이스 수가 주어진다.

다음으로 6자리의 숫자가 주어진다.

[입력예시]

3

667767

054060

101123

[출력예시]

#1 1

#2 1

#3 0

[A반] 4회차 월말평가 – 알고리즘 응용



| 문제 2 : 이진 탐색 (배점 30점)

오름차순으로 정렬된 N개의 숫자 목록과 이 목록에서 탐색할 M개의 숫자가 주어질 때, 이진 탐색 횟수가 가장 적은 숫자와 이 때의 탐색 횟수를 출력하려고 합니다.

- 이진 탐색 구간의 시작과 끝 인덱스를 l과 r이라고 하면, 중심 원소의 인덱스 $m = (l+r) // 2$ 이고, 이진 탐색의 왼쪽 구간은 l부터 m-1, 오른쪽 구간은 m+1부터 r이 됩니다.
- 탐색 횟수는 탐색 구간의 중심 원소를 탐색중인 정수와 비교하는 횟수입니다.
- **최소 탐색 횟수가 같은 숫자가 2개 이상 있는 경우 가장 작은 값을 출력합니다.**

정수 목록의 숫자는 중복되지 않으며, 탐색에 실패한 숫자는 탐색 횟수 비교에서 제외합니다. 최소한 1개의 숫자는 탐색에 성공합니다.

[A반] 4회차 월말평가 – 알고리즘 응용



[입력]

첫 번째 줄에 테스트 케이스의 수가 주어집니다. 각 테스트 케이스는 다음과 같이 구성되어 있습니다.

첫 번째 줄에는 순서대로,

1. 숫자 목록의 길이 N ($1 \leq N \leq 1024$)
2. 탐색 횟수를 비교할 숫자의 개수 M ($1 \leq M \leq 100$) 이 주어집니다.

두 번째 줄에는 탐색할 숫자인 M 개의 정수가 공백으로 구분되어 주어집니다.

세 번째 줄에는 정렬된 숫자 목록인 N 개의 정수가 공백으로 구분으로 주어집니다.

[출력]

각 테스트 케이스마다 #과 테스트케이스번호, 가장 탐색 횟수가 적은 숫자와 해당 숫자의 탐색 횟수를 띄어쓰기로 구분하여 출력합니다.

최소 탐색횟수가 같은 숫자가 2개 이상인 경우, 더 작은 숫자와 탐색횟수를 출력합니다.

입력 예시

```
3
3 2
1 2
1 2 3
5 3
0 1 2
1 2 3 4 5
10 3
1 3 4
1 2 3 4 5 6 7 8 9 10
```

출력 예시

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

[A반] 4회차 월말평가 – 알고리즘 응용

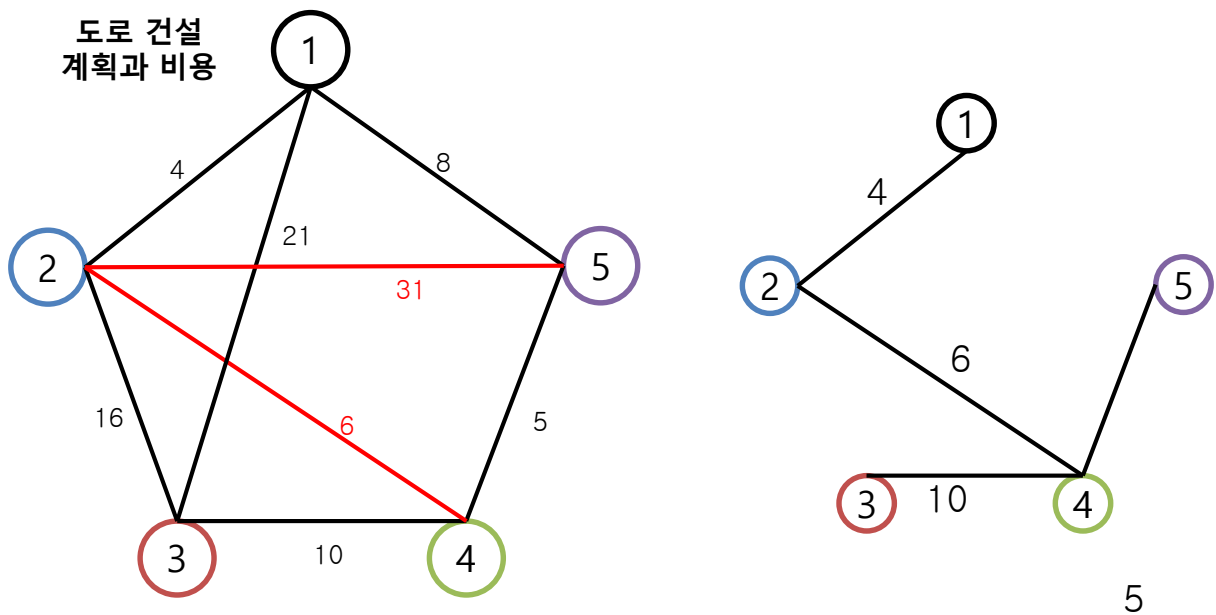


| 문제 3 : 도로 건설 (배점 30점)

싸피국은 V 개의 도시를 연결하는 E 개의 새로운 도로를 만들려고 합니다. 그러나 예산이 한정되어 있으므로, 새 도로만으로 한 도시에서 어떤 도시로도 이동할 수 있다면 계획된 모든 도로가 건설될 필요는 없습니다. (다른 도시를 경유해 이동 할 수 있습니다.) 꼭 필요한 도로만 최소한의 비용으로 만드는 것이 플랜 A입니다.

새로운 도로 계획이 주어 질 때, 플랜 A의 최소 건설 비용을 계산해 얼마의 예산이 남는지 출력하는 프로그램을 만드세요. 단, 비용이 예산을 초과하는 경우 플랜 B로 넘어갑니다.

다음 왼쪽 그림은 도로 건설 계획의 예로 도로 양끝의 도시와 필요한 비용을 알 수 있습니다. (도시마다 이름이 있지만 편의상 1번부터 V 번으로 사용합니다.)



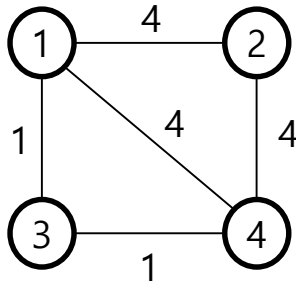
도로의 색은 겹치는 도로와 구분을 위함 이지 특별한 의미는 없습니다.

오른쪽 그림은 플랜 A를 실행했을 때의 결과로 비용은 25입니다. 예산이 30이라면 5가 남으므로 이를 출력합니다.

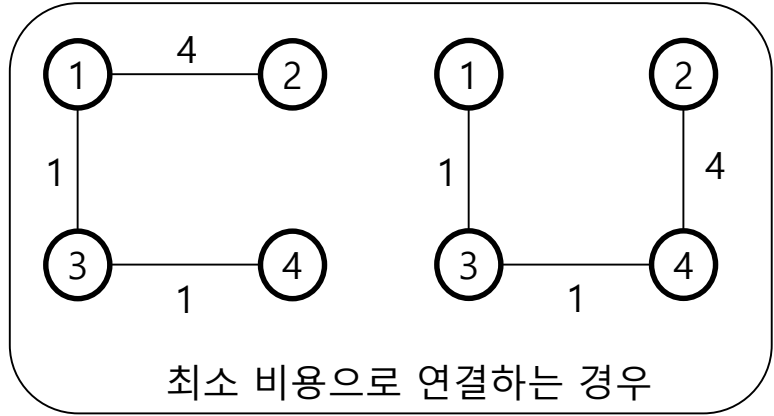
[A반] 4회차 월말평가 – 알고리즘 응용



다음 그림은 모든 도시를 연결하는 최소 건설 비용이 6인 경우 입니다. 이때 예산이 5라면 공사비가 예산을 초과하므로 플랜 B로 넘어갑니다.

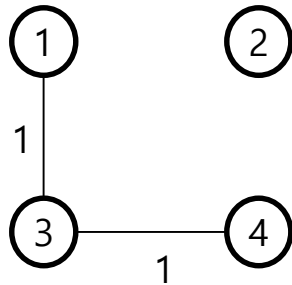


전체 계획



최소 비용으로 연결하는 경우

플랜 B는 모든 도시를 연결하는 대신 1에서 V번까지 이동 할 수 있는 도로만 최소 비용으로 만드는 것입니다. 이때의 비용이 예산을 초과하지 않으면 플랜 B가 성공한 것으로 남는 예산인 3을 출력합니다.



만약 플랜 A와 플랜 B 모두 예산을 초과하는 경우에는 -1을 출력합니다.

[A반] 4회차 월말평가 – 알고리즘 응용



입력

첫 줄에는 테스트 케이스의 개수 T 가 주어집니다. ($1 \leq T \leq 10$)

다음 줄부터, 각 테스트 케이스 별로 첫 줄에 도시의 수 V , 도로의 수 E , 건설예산 M , 둘째 줄 부터 도로 양끝의 도시 번호와 건설 비용이 E 개의 줄에 걸쳐 주어집니다.

$$4 \leq V \leq 100$$

$$V-1 \leq E \leq V*(V-1)/2$$

$$1 \leq M \leq 1000$$

출력

#과 테스트케이스 번호에 이어 빈칸으로 구분 후 남은 예산 혹은 -1을 출력합니다.

[A반] 4회차 월말평가 – 알고리즘 응용



입력 예시

출력 예시

```

3
5 8 30      # V E M
1 2 4      # 도시, 도시, 비용
1 3 21
1 5 8
2 3 16
2 4 6
2 5 31
3 4 10
4 5 5      # 여기까지 첫 번째 테스트 케이스
5 8 20
1 2 4
1 3 21
1 5 8
2 3 16
2 4 6
2 5 31
3 4 10
4 5 5
8 16 80
1 3 6
3 5 27
5 7 1
7 8 26
8 6 14
6 4 9
4 2 7
1 2 3
1 7 30
2 8 50
2 5 45
4 7 42
3 8 21
1 6 12
3 4 3
5 6 37
    
```

```

#1 5
#2 12
#3 18
    
```


[A반] 4회차 월말평가 – 알고리즘 응용



| 문제 4 : 재귀 (배점 : 10점)

```
1 def solution(n, k):
2     if n == k:
3         print(*nums)
4     else:
5         for i in range(n, k):
6             nums[n], nums[i] = nums[i], nums[n]
7             solution(n+1, k)
8             nums[n], nums[i] = nums[i], nums[n]
9
10 nums = [1, 2, 3]
11 solution(0, 3)
```

- 1) 위의 코드는 어떠한 동작을 하는 알고리즘인가?
- 2) 2번, 6번, 7번 라인이 의미하는 바를 간략하게 설명하시오.
 - 2번 라인
 - 6번 라인
 - 7번 라인
- 3) 위의 코드의 실행 결과를 적으시오.