

Homework #4

Due: 12/15

HW4 instructions

HW4 는 5 개의 coding 실습 문항으로 이루어져 있습니다. 각 문항별 Python 파일을 .zip 으로 압축하여 제출하여 주세요. 모든 코드는 Python(.py extension)으로 작성하시기 바랍니다.

형식: (HW4_학번_문제번호.py) >> HW4_학번_이름.zip 압축 후 제출

(예 : HW4_202200000_홍길동.zip)

[Greedy]

1. CPU 스케줄링 (15 points)

CPU 스케줄링은 멀티 프로그래밍을 가능하게 하는 운영체제의 동작 기법이다.

이러한 스케줄링에서 CPU 는 단위 시간(1 unit of time)에 실행(Running) 혹은 유휴(Idle) 상태를 가진다. 주어진 배열(L)은 현재 운영체제가 수행하고자 하는 n 개의 작업 목록인데, 이들의 순서는 정해져 있지 않다. 단, 같은 작업을 수행할 때는 m 의 휴식시간이 필요하다. 가령 $L = ['A', 'A', 'A', 'B', 'B', 'B']$ 에 대해 $m=2$ 인 상황에서 주어진 m 에 대해 가장 빨리 작업을 마치기 위한 스케줄링 방식은 A-B-Idle-A-B-Idle-A-B 로, 8 단위시간이 소요된다. 주어진 작업 목록에 대해 스케줄링을 완료하기까지 걸리는 최소 시간을 구하시오.

입력 : 첫째줄에 n 과 m 이 주어진다. ($1 \leq n \leq 10000$, $0 \leq m \leq 100$)

둘째줄에는 n 개의 작업목록이 주어진다.

출력 : 작업들이 가장 빨리 끝날수 있는 최소 시간을 출력한다.

예제 입력 1

6 2
AAABBB

예제 출력 1

8

예제 입력 2

6 0
AAABBB

예제 출력 2

6

예제 입력 3

12 2
AAAAAABCDEFG

예제 출력 3

16

2. 카카오 서버 (20 points)

카카오는 서버에 유저들의 데이터를 저장하기 위해 여러 디스크를 사용해 데이터를 분산 저장하고 있다. 카카오 엔지니어인 경택이는 효율적인 파일 시스템을 개발해 각각의 디스크들을 재포맷(Reformat) 하려고 한다. 재포맷된 디스크들은 재포맷되기 이전의 용량보다 커지거나 작아질 수 있다. 하지만 이 디스크들은 사용중이기 때문에, 함부로 포맷시킬 수는 없다. 모든 데이터들은 재포맷 되기전 다른 디스크 혹은 여유 디스크에 옮겨져야 한다. (재포맷 후 원래 디스크로 다시 옮겨놓을 필요는 없다.) 예를들어 4 개의 디스크 A,B,C,D 에 대해 기존의 용량이 6,1,3,3 GB 이고, 새로운 파일시스템을 적용한 재포맷 이후의 용량이 6,7,5,5 라고 한다면 1GB 의 여유 디스크만 필요하다. (B 를 먼저 포맷한다면 7G 의 용량이 생기기 때문에 A 의 데이터를 B 로 옮긴 다음 A 재포맷, A 의 6G 를 사용해 C 와 D 를 A 로 옮긴 다음 C,D 재포맷 순서로 해결하면 된다. 따라서 B 의 재포맷을 위한 1G 만 필요하다.) 이와 같이 모든 디스크 들을 재포맷 하기위한 여유디스크 의 최소 용량은 얼마가 필요한가?

입력 : 첫째 줄에 디스크 개수 n 이 주어진다. ($1 \leq n \leq 1000000$)

둘째 줄부터 각각의 디스크에 대해 재포맷 전후의 용량이 주어진다.

출력 : 모든 디스크들을 재포맷 하기위한 여유 디스크의 최소 용량을 출력한다.

예제 입력 1

```
4
6 6
1 7
3 5
3 5
```

예제 출력 1

1

예제 입력 2

4

2 2

3 3

5 1

5 10

예제 출력 2

5

[Network flow]

3. Minimum s-t cuts (15 points)

n 개의 정점과 m 개의 간선을 가지는 무방향 그래프 G 에 대해 m 개의 간선에 대한 가중치 정보가 주어졌을 때, 정점 s 와 t 를 분리시키는 최소 컷(s 와 t 를 분리시킬 때 지나가는 간선들의 가중치 합) 을 구하시오.

입력 : 첫째 줄에 정점의 개수 n , 간선 수 m 이 주어진다. ($2 \leq n \leq 500$, $1 \leq m \leq 10,000$)

둘째 줄부터 간선 m 개에 대한 연결정보(v_1, v_2, weight) 가 주어진다.

($1 \leq v_1, v_2 \leq n$, $1 \leq \text{weight} \leq 100$, $v_1 \neq v_2$)

마지막 줄에는 s, t 가 각각 주어진다.

예제 입력 1

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

예제 출력 1

6

4. 숨은 고수 (20 points)

소현이는 2 학년 6 반 스타크래프트 숨은 고수이다. 여름방학을 맞이해 반에서 스타크래프트 리그가 열렸는데, 방학동안 학생들이 1 대 1 로 여러 번 대결한다. 게임은 무조건 승패가 나며 이긴사람이 승점 1 점을 얻게 된다. 모든 게임이 진행된 후 1 등은 승점이 가장 많은 학생으로 결정되며 승점이 같을시 공동 1 등이 존재할 수 있다. 소현이는 숨은 실력을 드러내지 않기 위해 최소한의 승점만 가지고 리그에서 단독 1 등을 하고 싶다. 이때까지 진행된 리그의 각 학생들의 승점과 남아있는 게임정보가 주어질 때, 소현이가 단독 1 등을 할 수 있는지 여부를 판단하고 만약 할 수 있다면 그때의 소현이의 승점을 구하는 프로그램을 작성하시오.

입력 : 첫째 줄에는 리그의 참가하는 학생 수 n 과 남아있는 경기수 m 이 주어진다. .
($2 \leq n \leq 12, 0 \leq m \leq 100$) 학생들은 0 부터 $n-1$ 까지 번호가 부여되는데, 소현이의 번호는 항상 0 번이다. 둘째 줄에는 각 학생별로 현재 승점이 주어진다. 셋째 줄부터는 남은 m 개의 경기에 대해 게임을 붙는 두 학생의 번호가 주어진다. 또한, 리그에서는 모든 학생이 동일한 수의 경기를 하진 않는다.

출력 : 소현이가 단독 1 위를 하기위한 최소 승점을 출력한다. 만약 단독 1 위가 불가능하다면, -1 을 출력한다.

예제 입력 1

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

예제 출력 1

```
5
```

예제 입력 2

```
3 3
4 2 2
1 2
1 2
1 2
```

예제 출력 2

```
-1
```

예제 입력 3

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

2 3

1 3

예제 출력 3

5

[NP Problem]

5. Travelling Salesman Problem (30 points : 정답점수 20points, 시간점수 10points)

실시간 시스템 연구실 소속 민기는 논문발표를 위해 학교에서 출발해 N 개의 도시를 방문하고 본인의 집으로 돌아오려 한다. 학교와 집, 각각의 도시의 위치는 좌표 (x, y) 로 주어지고 $(0 \leq x \leq 100, 0 \leq y \leq 100)$ 두 위치 $(x_1, y_1), (x_2, y_2)$ 에 대해 맨하탄 거리(Manhattan Distance)를 따른다. $(|x_1 - x_2| + |y_1 - y_2|)$

민기는 전용 비행기의 연료 제한으로 N ($2 \leq N \leq 20$)개의 도시를 모두 방문하고 집으로 돌아오는 경로 중 최대한 짧은 경로로 이동하고자 한다. 각 도시의 위치가 주어질 때, 민기가 이동하기 위한 최단 경로를 구하는 프로그램을 작성하시오.

입력 : 첫째 줄에는 도시의 수 N 이 주어진다.

두 번째 줄부터는 N 개 도시의 위치좌표 (x, y) 가 주어진다. 회사, 집을 포함한 모든 좌표는 서로 다른 위치에 존재한다.

출력 : 학교에서 집까지의 최단경로를 출력한다.

예제 입력 1

```
5
0 0 100 100 70 40 30 10 10 5 90 70 50 20
```

예제 출력 1

```
200
```

예제 입력 2

```
6
88 81 85 80 19 22 31 15 27 29 30 10 20 26 5 14
```

예제 출력 2

```
304
```

예제 입력 3

```
12
0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 12 12 13 13
```

예제 출력 3

```
50
```