

금과 은 운반하기

Python3

문제 설명

어느 왕국에 하나 이상의 도시들이 있습니다. 왕국의 왕은 새 도시를 짓기로 결정하였습니다. 해당 도시를 짓기 위해서는 도시를 짓는 장소에 금 a kg과 은 b kg이 전달되어야 합니다.

각 도시에는 번호가 매겨져 있는데, i 번 도시에는 금 $g[i]$ kg, 은 $s[i]$ kg, 그리고 트럭 한 대가 있습니다. i 번 도시의 트럭은 오직 새 도시를 짓는 건설 장소와 i 번 도시만을 왕복할 수 있으며, 편도로 이동하는 데 $t[i]$ 시간이 걸리고, 최대 $w[i]$ kg 광물을 운반할 수 있습니다. (광물은 금과 은입니다. 즉, 금과 은을 동시에 운반할 수 있습니다.) 모든 트럭은 같은 도로를 여러 번 왕복할 수 있으며 연료는 무한대라고 가정합니다.

정수 a , b 와 정수 배열 g , s , w , t 가 매개변수로 주어집니다. 주어진 정보를 바탕으로 각 도시의 트럭을 최적으로 운행했을 때, 새로운 도시를 건설하기 위해 금 a kg과 은 b kg을 전달할 수 있는 가장 빠른 시간을 구해 return 하도록 solution 함수를 완성해주세요.

제한사항

- $0 \leq a, b \leq 10^9$
- $1 \leq g$ 의 길이 = s 의 길이 = w 의 길이 = t 의 길이 = 도시 개수 $\leq 10^5$
 - $0 \leq g[i], s[i] \leq 10^9$
 - $1 \leq w[i] \leq 10^2$
 - $1 \leq t[i] \leq 10^5$
 - $a \leq g$ 의 모든 수의 합
 - $b \leq s$ 의 모든 수의 합

입출력 예

a	b	g	s	w	t	result
10	10	[100]	[100]	[7]	[10]	50
90	500	[70, 70, 0]	[0, 0, 500]	[100, 100, 2]	[4, 8, 1]	499

입출력 예 설명

입출력 예 #1

- 도시가 오직 하나뿐이므로, 0번 도시의 유일한 트럭이 모든 운반을 해결해야 합니다. 이 트럭은 최대 7kg만큼의 광물을 운반할 수 있으며 편도 완주에는 10시간이 걸립니다.
- 맨 처음에 10시간을 써서 7kg만큼의 금을 운반하고, 10시간을 써서 다시 도시로 돌아오고, 10시간을 써서 7kg만큼의 은을 운반하고, 10시간을 써서 다시 도시로 돌아오고, 마지막으로 10시간을 써서 3kg만큼의 금과 3kg만큼의 은을 운반하면, 총 50시간 만에 필요한 모든 금과 은을 조달할 수 있습니다.
- 따라서, 50을 return 해야 합니다.

입출력 예 #2

- 도시가 3개이고, 0번과 1번 도시는 금만 70kg씩 가지고 있고 2번 도시는 은을 500kg 가지고 있습니다.
 - 0번 도시의 트럭은 용량은 100kg, 편도 완주 시간은 4시간입니다.
 - 1번 도시의 트럭은 용량은 100kg, 편도 완주 시간은 8시간입니다.
 - 2번 도시의 트럭은 용량은 2kg, 편도 완주 시간은 1시간입니다.
- 금은 0번 도시의 트럭과 1번 도시의 트럭이 각각 45kg씩 나누어서 운반하면 8시간 안에 필요한 모든 금을 조달할 수 있습니다.

- 은은 2번 도시의 트럭이 한 번에 2kg씩 250번 운반하면(249번 왕복 + 1번 편도) 총 499시간 만에 필요한 모든 은을 조달할 수 있습니다.
- 따라서, 499를 return 해야 합니다.

solution.py

```
1 def solution(a, b, g, s, w, t):
2     ## a 필요한 금
3     ## b 필요한 은
4     ## g 갖고있는 금
5     ## s 갖고있는 은
6     ## t 편도 이동 시간
7     ## w 운반할 수 있는 무게
8
9     ## 새로운 도시를 건설하기 위해 금과 은을 전달 할 수 있는 가장빠른시간을 return 하세요.
10
11
12     ## 시간대비 옮길 수 있는 양 리스트를 만들
13     wfort = []
14     for idx,(w,t) in enumerate(zip(w,t)):
15         wfort.append([idx,w/t])
16     wfort.sort(key = lambda x:-x[1])
17     print(wfort)
18
19     time = 0
20     while(a >= 0 or b >= 0):
21         for idx,wt in wfort:
22             ## 금
23             ## 만약 재고가 남아있으면
24             if g[idx] - w[idx] >= 0:
25                 g[idx] -= w[idx]
26                 a -= w[idx]
27                 time += 2*t[idx]
28                 break
29         for idx,wt in wfort:
30             ## 은
31             ## 만약 재고가 남아있으면
32             if s[idx] - w[idx] >= 0:
33                 s[idx] -= w[idx]
34                 b -= w[idx]
35                 time += 2*t[idx]
36                 break
37     print(a,b,time)
```

실행 결과

실행 결과가 여기에 표시됩니다.

초기화

코드 실행

제출 후 채점하기