

동전 0

[1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 50000]

← i ↑

4200

n → 0 Stop

$$4200 // 1000 \rightarrow 4$$

$$200 // 500 \rightarrow 0$$

$$200 // 100 \rightarrow 2$$

If N이 나누어지면

동전 개수 +

N -

else:

Continue.

손해시간 = 회의실 빈 시간 + 회의 경과 시간

$$(\text{시작}, \text{끝}) = (x, y) \rightarrow \text{손해시간} = x - \text{cur_time} + y - x$$

$$= \underbrace{y}_{\downarrow} - \underbrace{\text{cur_time}}_{\text{고정값}}$$

최소 \Rightarrow 덜 손해 \Rightarrow 많은 회의

$\therefore y$ 기준 정렬

ex) $\begin{matrix} 2 & 2 \\ 1 & 2 \end{matrix} \Rightarrow$ 예외사항, $\therefore x$ 기준 정렬

N 명 , $P_i = i$ 번 사람의 인출 시간

$$p = [3, 1, 4, 3, 2]$$

* $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ *

$$3 + (3+1) + (3+1+4) + (3+1+4+3) + (3+1+4+3+2) = 39 \text{ 분}$$

$$3 + 4 + 8 + 11 + 13$$

* $2 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 3$

$$1 + (1+2) + (1+2+3) + (1+2+3+3) + (1+2+3+3+4) = 32 \text{ 분}$$

$$1 + 3 + 6 + 9 + 13$$

누적합 $\downarrow \Rightarrow$ 최소 시간 우선

$$\begin{matrix} i & 1 & 2 & 3 & 4 & 5 \\ p = [& 3, & 1, & 4, & 3, & 2 \end{matrix}]$$

$$p.sort = [1, 2, 3, 3, 4]$$

$$1 + (1+2) + (1+2+3) + (1+2+3+3) + (1+2+3+3+4)$$

$$dp = [p[1], p[2] + dp[1], p[3] + dp[2], p[4] + dp[3] \dots]$$

$$dp[i] = p[i] + dp[i-1]$$

$$\text{sum}(dp)$$

양수, +, -, ()

$$55 - (50 + 40) = -35$$

[55] +

[50, 40] -

Input : 1+2-3+4+5-6+7+8

+ [1, 2]

사실상, 최초의 -가 반전되면

- [3, 4, 5, 6, 7, 8]

그냥 다 - 리스트에 집어넣어서 상관x

$$TL : sum(+) - sum(-)$$

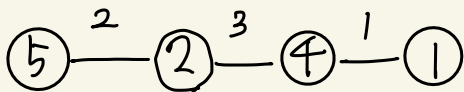
Input : $1+2-3+4+5-6+7+8$

+ [1, 2]

사실상, 최초의 -가 바뀌면

- [3, 4, 5, 6, 7, 8]

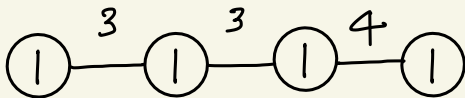
그냥 다 - 리스트에 집어넣어서 상관x



6L \longrightarrow 30원

2L 3L 1L \longrightarrow 20원

2L 4L \longrightarrow 18원



10L \longrightarrow 10원

#1. 첫 번째 도시에서는 반드시 주유를 해야함.

#2. 최소의 가격으로 많은 거리를 이동해야한다.

Price = [5, 2, 4, 1]

distance = [2, 3, 1]

min = 5 \rightarrow #1, ans = min * dist[0]

for i : 1 ~ len(distance):

if Price[i] < min :

min = Price[i]

ans += min * distance[i]

"제일싼 주유소 → 끝"

"두 번째로싼 주유소 → 제일싼"

"세 번째로 → 두 번째"

input [5, 2, 4, 1] distance [2, 3, 1]

→ [1, 2, 4, 5]

Price = [8,

While "input 리스트 안에 아무것도 없는 때까지"

제일싼 주유소

if (input[0]).index == -1:

min = input[1]

else:

min = input[0]

Price.append(min * sum(distance[min.index:]))

반복을 위한
update → input = input[:min.index].sort()