

# ATM 11399

2022년 02월 16일

PEPSI

발표자: 최건희

# 문제

인하은행에는 ATM이 1대밖에 없다. 지금 이 ATM앞에  $N$ 명의 사람들이 줄을 서있다. 사람은 1번부터  $N$ 번까지 번호가 매겨져 있으며,  $i$ 번 사람이 돈을 인출하는데 걸리는 시간은  $P_i$ 분이다.

사람들이 줄을 서는 순서에 따라서, 돈을 인출하는데 필요한 시간의 합이 달라지게 된다. 예를 들어, 총 5명이 있고,  $P_1 = 3, P_2 = 1, P_3 = 4, P_4 = 3, P_5 = 2$  인 경우를 생각해 보자. [1, 2, 3, 4, 5] 순서로 줄을 선다면, 1번 사람은 3분만에 돈을 뽑을 수 있다. 2번 사람은 1번 사람이 돈을 뽑을 때 까지 기다려야 하기 때문에,  $3+1 = 4$ 분이 걸리게 된다. 3번 사람은 1번, 2번 사람이 돈을 뽑을 때까지 기다려야 하기 때문에, 총  $3+1+4 = 8$ 분이 필요하게 된다. 4번 사람은  $3+1+4+3 = 11$ 분, 5번 사람은  $3+1+4+3+2 = 13$ 분이 걸리게 된다. 이 경우에 각 사람이 돈을 인출하는데 필요한 시간의 합은  $3+4+8+11+13 = 39$ 분이 된다.

줄을 [2, 5, 1, 4, 3] 순서로 줄을 서면, 2번 사람은 1분만에, 5번 사람은  $1+2 = 3$ 분, 1번 사람은  $1+2+3 = 6$ 분, 4번 사람은  $1+2+3+3 = 9$ 분, 3번 사람은  $1+2+3+3+4 = 13$ 분이 걸리게 된다. 각 사람이 돈을 인출하는데 필요한 시간의 합은  $1+3+6+9+13 = 32$ 분이다. 이 방법보다 더 필요한 시간의 합을 최소로 만들 수는 없다.

줄을 서 있는 사람의 수  $N$ 과 각 사람이 돈을 인출하는데 걸리는 시간  $P_i$ 가 주어졌을 때, 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 구하는 프로그램을 작성하시오.

## 입력

첫째 줄에 사람의 수  $N$  ( $1 \leq N \leq 1,000$ )이 주어진다. 둘째 줄에는 각 사람이 돈을 인출하는데 걸리는 시간  $P_i$ 가 주어진다. ( $1 \leq P_i \leq 1,000$ )

## 출력

첫째 줄에 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 출력한다.

# 문제

---

## ◆ 문제 해석

$$✓ A = p_1$$

$$✓ B = p_1 + p_2$$

$$✓ C = p_1 + p_2 + p_3$$

$$\text{총 합: } A + B + C$$

# 문제

## ◆ 문제 풀이 전략

- 소요시간이 작은 사람을 먼저 뽑게 해준다. 따라서 입력되는  $p_i$  를 오름차순으로 정렬하고, 누적되는 소요시간을 계산해준다.

## ◆ 타당성

- 먼저 오는 수가 가장 많이 더해지기 때문에, 최소 소요시간을 위해서는 먼저 오는 수 일수록 작은 수가 되어야 한다.

$$\begin{aligned}
 &\checkmark A = p_1 \\
 &\checkmark B = p_1 + p_2 \\
 &\checkmark C = p_1 + p_2 + p_3
 \end{aligned}$$

총 합:  $A + B + C$

# 풀이

## ◆ CODE

```
int main()
{
    int a = 0;
    int time[1000] = {0,};
    scanf("%d",&a);
    for(int i=0;i<a;i++)
    {
        scanf("%d",&time[i]);
    }
}
```

정렬 해주기

(어떤 정렬 알고리즘을 선택해야 할까? 근거는?)

```
int ans = 0;
int buf = 0;
```

누적 된 소요시간 계산해주기

```
printf("%d",ans);
```

# 풀이

## ◆ CODE

```
int main()
{
    int a = 0;
    int time[1000] = {0,};
    scanf("%d",&a);
    for(int i=0;i<a;i++)
    {
        scanf("%d",&time[i]);
    }

    MergeSort(time,0,a-1);  ✓  worst case:  $O(n \log n)$ 

    int ans = 0;
    int buf = 0;

    누적 된 소요시간 계산해주기

    printf("%d",ans);
}
```

# 풀이

## ◆ CODE

```
int main()
{
    int a = 0;
    int time[1000] = {0,};
    scanf("%d",&a);
    for(int i=0;i<a;i++)
    {
        scanf("%d",&time[i]);
    }

    MergeSort(time,0,a-1);  ✓ worst case:  $O(n \log n)$ 

    int ans = 0;
    int buf = 0;
    for(int i=0;i<a;i++)
    {
        buf += time[i];
        ans += buf;
    }

    printf("%d",ans);
}
```

# 🔍 Question?

---

