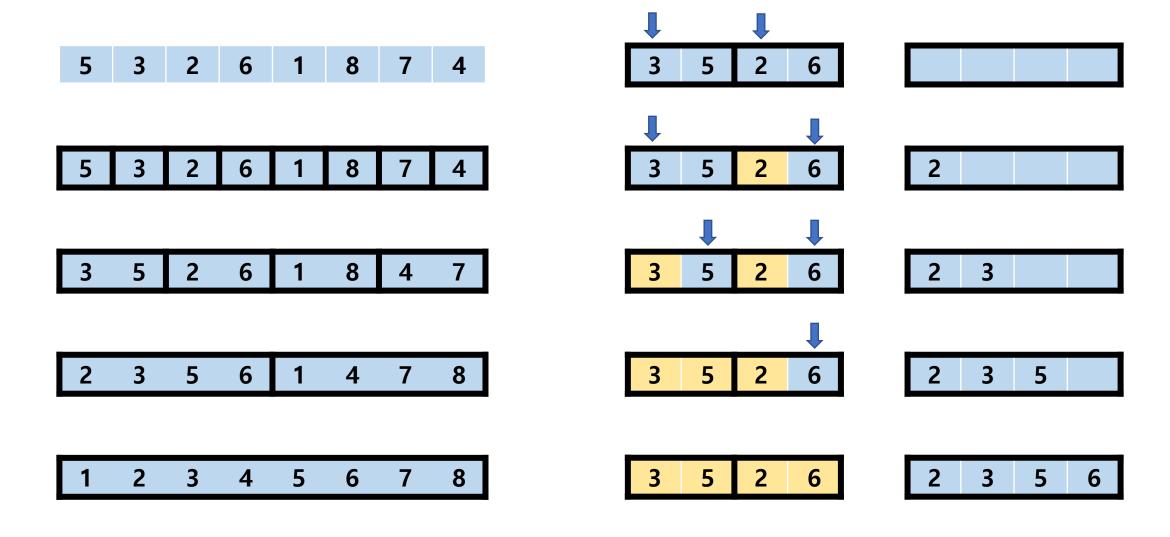
알고리즘 기초 세미나

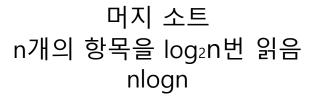
최재혁

200508

머지 소트



머지 소트, 왜 빠를까?









버블 소트 최대 1+2+3+...+n번 읽음 n(n+1)/2

2	2	E	1	6	7	Λ	Q
3	_	.	ı	U		4	0

머지 소트

```
1 void merge(int *arr, int l, int m, int r)
         int i = 0, j = 0, k = 1;
         int L[n1], R[n2];
         for (int idx = 0; idx < n1; idx++)
              L[idx] = arr[l + idx];
         for (int idx = 0; idx < n2; idx++)
    R[idx] = arr[m + 1 + idx];</pre>
12
13
14
         while (i < n1 && j < n2)
              if (L[i] \leftarrow R[j])
                  arr[k] = L[i];
                   i++;
              }
else
20
21
22
23
24
25
26
27
28
30
31
32
33
34
35
36
37
38
                   arr[k] = R[j];
                   j++;
              k++;
         while (i < n1)
              arr[k] = L[i];
              i++;
              k++;
         while (j < n2)
              arr[k] = R[j];
              j++;
k++;
40 }
42 void merge_sort(int arr[], int l, int r)
43 {
         if (l < r)
              int m = l + (r - l) / 2;
              merge_sort(arr, l, m);
49
50
51
52
53 }
              merge_sort(arr, m + 1, r);
              merge(arr, l, m, r);
```



풀이 (2751.cpp)

```
1 #include <iostream>
 2 using namespace std;
 4 void merge_sort(int arr[], int l, int r)
 5 {
6
7 }
8
 9 int main()
10 {
        int n; cin >> n;
11
12
        int arr[n];
        for(int i = 0; i < n; i++)
    cin >> arr[i];
13
14
15
16
        merge_sort(arr, 0, n-1);
17
        for (int i = 0; i < n; i++)
    cout << arr[i] << "\n";</pre>
18
19
20
21
        return 0;
22 }
```

11399번, ATM (<u>www.acmicpc.net/problem/11399</u>)



문제

인하은행에는 ATM이 1대밖에 없다. 지금 이 ATM앞에 N명의 사람들이 줄을 서있다. 사람은 1번부터 N번까지 번호가 매겨져 있으며, i번 사람이 돈을 인출하는데 걸리는 시간은 P.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 생각해보자. [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=4분이 걸리게 된다. 3번 사람은 1번, 2번 사람이 돈을 뽑을 때까지 기다려야 하기 때문에, 총 3+1+4=8분이 필요하게 된다. 4번 사람은 3+1+4+3=11분, 5번 사람은 3+1+4+3+2=13분이 걸리게 된다. 이 경우에 각 사람이 돈을 인출하는데 필요한 시간의 합은 3+4+8+11+13=39분이 된다.

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

줄을 서 있는 사람의 수 N과 각 사람이 돈을 인출하는데 걸리는 시간 P_i가 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 한의 최솟값을 출력한다.

풀이 (11399.cpp)

```
1 #include <iostream>
 2 #include <algorithm>
 3 using namespace std;
 5 int main()
 6 { 7
       int n;
       scanf("%d", &n);
       int arr[n], sum = 0, ret = 0;
       for (int i = 0; i < n; i++)
10
           cin >> arr[i];
11
12
13
       sort(arr, arr + n);
14
       for (int i = 0; i < n; i++)
15
16
           sum += arr[i];
17
           ret += sum;
       }
18
19
20
       cout << ret;</pre>
21
22
       return 0;
23 }
```

QnA