

알고리즘 기초 세미나

최재혁

200508

머지 소트

5	3	2	6	1	8	7	4
---	---	---	---	---	---	---	---

5	3	2	6	1	8	7	4
---	---	---	---	---	---	---	---

3	5	2	6	1	8	4	7
---	---	---	---	---	---	---	---

2	3	5	6	1	4	7	8
---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

3	5	2	6
---	---	---	---

3	5	2	6
---	---	---	---

3	5	2	6
---	---	---	---

3	5	2	6
---	---	---	---

3	5	2	6
---	---	---	---

--	--	--	--

2			
---	--	--	--

2	3		
---	---	--	--

2	3	5	
---	---	---	--

2	3	5	6
---	---	---	---

머지 소트, 왜 빠를까?

머지 소트
n개의 항목을 $\log_2 n$ 번 읽음
 $n \log n$



버블 소트
최대 $1+2+3+\dots+n$ 번 읽음
 $n(n+1)/2$



머지 소트

```
1 void merge(int *arr, int l, int m, int r)
2 {
3     int i = 0, j = 0, k = l;
4     int n1 = m - l + 1, n2 = r - m;
5     int L[n1], R[n2];
6
7     for (int idx = 0; idx < n1; idx++)
8         L[idx] = arr[l + idx];
9     for (int idx = 0; idx < n2; idx++)
10        R[idx] = arr[m + 1 + idx];
11
12    while (i < n1 && j < n2)
13    {
14        if (L[i] <= R[j])
15        {
16            arr[k] = L[i];
17            i++;
18        }
19        else
20        {
21            arr[k] = R[j];
22            j++;
23        }
24        k++;
25    }
26
27    while (i < n1)
28    {
29        arr[k] = L[i];
30        i++;
31        k++;
32    }
33
34    while (j < n2)
35    {
36        arr[k] = R[j];
37        j++;
38        k++;
39    }
40 }
41
42 void merge_sort(int arr[], int l, int r)
43 {
44     if (l < r)
45     {
46         int m = l + (r - l) / 2;
47
48         merge_sort(arr, l, m);
49         merge_sort(arr, m + 1, r);
50
51         merge(arr, l, m, r);
52     }
53 }
```

2751번, 수 정렬하기 2 (www.acmicpc.net/problem/2751)

수 정렬하기 2 성공



Silver V

정렬

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	256 MB	74470	21625	13861	31.535%

문제

N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

입력

첫째 줄에 수의 개수 N ($1 \leq N \leq 1,000,000$)이 주어진다. 둘째 줄부터 N개의 줄에는 숫자가 주어진다. 이 수는 절댓값이 1,000,000보다 작거나 같은 정수이다. 수는 중복되지 않는다.

출력

첫째 줄부터 N개의 줄에 오름차순으로 정렬한 결과를 한 줄에 하나씩 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
1
2
3
4
5
```

풀이 (2751.cpp)

```
1 #include <iostream>
2 using namespace std;
3
4 void merge_sort(int arr[], int l, int r)
5 {
6     /* sort */
7 }
8
9 int main()
10 {
11     int n; cin >> n;
12     int arr[n];
13     for(int i = 0; i < n; i++)
14         cin >> arr[i];
15
16     merge_sort(arr, 0, n-1);
17
18     for (int i = 0; i < n; i++)
19         cout << arr[i] << "\n";
20
21     return 0;
22 }
```

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

ATM

상공 분류



Silver III

탐욕법 정렬

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	24554	15922	13493	66.139%

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 최소값을 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 최소값을 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
32
```

풀이 (11399.cpp)

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 int main()
6 {
7     int n;
8     scanf("%d", &n);
9     int arr[n], sum = 0, ret = 0;
10    for (int i = 0; i < n; i++)
11        cin >> arr[i];
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;
21
22    return 0;
23 }
```


QnA