

백준 1715번 - 카드 정렬하기[문제](#)[입력](#)[출력](#)[예제 입력1](#)[예제 출력1](#)[출처](#)[알고리즘 분류](#)[접근 방법](#)[소스코드](#)

백준 1715번 - 카드 정렬하기

시간제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2초	128MB	21083	7277	5732	35.092%

문제

정렬된 두 묶음의 숫자 카드가 있다고 하자. 각 묶음의 카드의 수를 A, B라 하면 보통 두 묶음을 합쳐서 하나로 만드는 데에는 A+B 번의 비교를 해야 한다. 이를테면, 20장의 숫자 카드 묶음과 30장의 숫자 카드 묶음을 합치려면 50번의 비교가 필요하다.

매우 많은 숫자 카드 묶음이 책상 위에 놓여 있다. 이들을 두 묶음씩 골라 서로 합쳐나간다면, 고르는 순서에 따라서 비교 횟수가 매우 달라진다. 예를 들어 10장, 20장, 40장의 묶음이 있다면 10장과 20장을 합친 뒤, 합친 30장 묶음과 40장을 합친다면 $(10 + 20) + (30 + 40) = 100$ 번의 비교가 필요하다. 그러나 10장과 40장을 합친 뒤, 합친 50장 묶음과 20장을 합친다면 $(10 + 40) + (50 + 20) = 120$ 번의 비교가 필요하므로 덜 효율적인 방법이다.

N개의 숫자 카드 묶음의 각각의 크기가 주어질 때, 최소한 몇 번의 비교가 필요한지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. ($1 \leq N \leq 100,000$) 이어서 N개의 줄에 걸쳐 숫자 카드 묶음의 각각의 크기가 주어진다. 숫자 카드 묶음의 크기는 1,000보다 작거나 같은 양의 정수이다.

출력

첫째 줄에 최소 비교 횟수를 출력한다.

예제 입력1

```
1 3
2 10
3 20
4 40
```

예제 출력1

1 | 100

출처

[출처](#)

알고리즘 분류

- 자료구조
- 그리디 알고리즘
- 우선순위 큐

접근 방법

카드 묶음을 순서대로 $s_1, s_2, s_3, \dots, s_n$ 으로 가정하고, 매 라운드에서 s_k, s_{k+1} 이 묶인다고 가정하자.

임의의 라운드에서 위와 같이 묶일 경우, 다음 라운드에서 총 비교 횟수는 어떻게 되는지 확인해보자.

Round 1 = p_1

총 비교 횟수

$$(s_1 + s_2) + (s_3 + s_4) + \dots + (s_{n-1} + s_n)$$

Round 2 = p_2

총 비교 횟수

$$\{(s_1 + s_2) + (s_3 + s_4) + \dots + (s_{n-1} + s_n)\} +$$

(1)

$$(s_1 + s_2 + s_3 + s_4) + (s_5 + s_6 + s_7 + s_8) + \dots + (s_{n-3} + s_{n-2} + s_{n-1} + s_n)$$

1번째 라운드에서 총 비교 횟수를 p_1 이라고 하였을 때 N 번째 라운드까지 진행하였을 때 p_1 은 N 번 반복하는 것을 알 수 있다. 마찬가지로 2번째 라운드에서 p_2 는 $N - 1$ 번 반복하는 것을 관찰할 수 있다.

즉, 각 라운드에서 필요한 총 비교 횟수 $p_k = Kp_1 + (K - 1)p_2 + \dots + p_k$ 임을 알 수 있다.

전체의 최적해를 얻기 위해선, $p_1 \leq p_2 \leq p_3 \leq \dots \leq p_n$ 를 유지하는 것이 중요하다.

이를 위해 각 라운드에서 선택할 수 있는 최소의 s_k, s_{k+1} 를 선택하는 것이 중요하다.

N 의 최대 값이 10^5 라는 것을 고려하였을 때 $O(\log N)$ 내에 최소 원소 2개를 선택하기 위해, Min heap을 이용하는 것이 핵심이다.

+ 소스코드 설명

아래 소스코드에서 while문 내부에서 최소값인 2개의 원소를 꺼내서 합한 다음, 1개의 원소를 삽입하여 loop를 진행하므로

`priority_queue` 내부의 원소의 개수는 linear하게 1씩 감소하는 것을 확인할 수 있다. 큐 내부에 원소가

$N(1 \leq N \leq 100,000)$ 인 것을 고려해보았을 때, 해당 알고리즘의 종료성을 확인할 수 있을 것이다.

소스코드

```
1  #define FASTIO cin.tie(0)->sync_with_stdio(false), cout.tie(0)
2  #include <bits/stdc++.h>
3  using namespace std;
4  int N;
5  int main(void){
6      FASTIO;
7      //////////////////////////////////////
8      cin >> N;
9      vector<long long> v(N);
10     for(auto &item : v)
11         cin >> item;
12
13     long long ans = 0;
14     priority_queue<long long, vector<long long>, greater<long long>> PQ;
15     for(const auto &item : v)
16         PQ.push(item);
17     while(PQ.size() > 1){
18         long long a, b;
19         a = PQ.top(), PQ.pop();
20         b = PQ.top(), PQ.pop();
21         ans += (a + b);
22         PQ.push(a + b);
23     }
24     cout << ans ;
25     return 0;
26 }
```