



#2470. 두 용액

<https://www.acmicpc.net/problem/2470>

25.04.21



Problem

시간: 40분

두 용액

성공 스페셜 저지

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초 (추가 시간 없음)	128 MB	73532	24283	17520	31.900%

문제

KOI 부설 과학연구소에서는 많은 종류의 산성 용액과 알칼리성 용액을 보유하고 있다. 각 용액에는 그 용액의 특성을 나타내는 하나의 정수가 주어졌다. 산성 용액의 특성값은 1부터 1,000,000,000까지의 양의 정수로 나타내고, 알칼리성 용액의 특성값은 -1부터 -1,000,000,000까지의 음의 정수로 나타낸다.

같은 양의 두 용액을 혼합한 용액의 특성값은 혼합에 사용된 각 용액의 특성값의 합으로 정의한다. 이 연구소에서는 같은 양의 두 용액을 혼합하여 특성값이 0에 가장 가까운 용액을 만들려고 한다.

예를 들어, 주어진 용액들의 특성값이 [-2, 4, -99, -1, 98]인 경우에는 특성값이 -99인 용액과 특성값이 98인 용액을 혼합하면 특성값이 -1인 용액을 만들 수 있고, 이 용액이 특성값이 0에 가장 가까운 용액이다. 참고로, 두 종류의 알칼리성 용액만으로나 혹은 두 종류의 산성 용액만으로 특성값이 0에 가장 가까운 혼합 용액을 만드는 경우도 존재할 수 있다.

산성 용액과 알칼리성 용액의 특성값이 주어졌을 때, 이 중 두 개의 서로 다른 용액을 혼합하여 특성값이 0에 가장 가까운 용액을 만들어내는 두 용액을 찾는 프로그램을 작성하시오.

입력

첫째 줄에는 전체 용액의 수 N 이 입력된다. N 은 2 이상 100,000 이하이다. 둘째 줄에는 용액의 특성값을 나타내는 N 개의 정수가 빈칸을 사이에 두고 주어진다. 이 수들은 모두 -1,000,000,000 이상 1,000,000,000 이하이다. N 개의 용액들의 특성값은 모두 다르고, 산성 용액만으로나 알칼리성 용액만으로 입력이 주어지는 경우도 있을 수 있다.

출력

첫째 줄에 특성값이 0에 가장 가까운 용액을 만들어내는 두 용액의 특성값을 출력한다. 출력해야 하는 두 용액은 특성값의 오름차순으로 출력한다. 특성값이 0에 가장 가까운 용액을 만들어내는 경우가 두 개 이상일 경우에는 그 중 아무것이나 하나를 출력한다.

예제 입력 1 복사

5
-2 4 -99 -1 98

예제 출력 1 복사

-99 98



Hint

Step 1. 문제 유형



1. 투 포인터

Two Pointer Algorithm이란?

정의

- 두 개의 포인터를 이용해 선형 탐색을 최적화하는 알고리즘
- 보통 정렬된 배열에서 구간이나 쌍(pair)을 찾는 문제에 사용

활용 예시

- 두 수의 합이 특정 값이 되는 쌍 찾기
- 구간 내 조건 만족하는 최댓값/최솟값
- 중복 없는 부분 문자열 길이



Hint

Step 1. 문제 유형



1. 투 포인터

이 문제에서 **Two Pointer**를 사용해야 하는 이유

브루트포스 vs 투 포인터

- 브루트포스: $O(N^2)$
- 투 포인터: $O(N)$ 또는 $O(N \log N)$ (정렬 + 투 포인터)

장점

- 선형 시간에 복잡한 조건 처리 가능
- 정렬된 데이터에서 매우 효율적

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초 (추가 시간 없음)	128 MB	73532	24283	17520	31.900%

입력

첫째 줄에는 전체 용액의 수 N 이 입력된다. N 은 2 이상 100,000 이하이다. 둘째 줄에는 용액의 특성값을 나타내는 N 개의 정수가 빈칸을 사이에 두고 주어진다. 이 수들은 모두 -1,000,000,000 이상 1,000,000,000 이하이다. N 개의 용액들의 특성값은 모두 다르고, 산성 용액만으론 알칼리성 용액만으로 입력이 주어지는 경우도 있을 수 있다.



Hint

Step 1. 문제 풀이 전략



1. 투 포인터

두 수의 합이 0에 가장 가까운 쌍

문제 조건

- N개의 수 중, 두 수를 골라 합이 0에 가장 가까운 쌍을 찾기

풀이 전략

1. 용액의 특성값을 담은 배열 정렬 오름차순으로 정렬
2. 목적은 절댓값이 0인 두 수를 찾는 것이므로 양쪽 끝에서 시작 (left, right)
3. 합이 0에 더 가까우면 정답 갱신
4. 합이 음수면 left++
5. 합이 양수면 right--
6. left와 right가 교차되면 탐색 종료



Hint

용액의 특성값을 오름차순으로 정렬했기 때문에 합이 음수인 상황에서 **right**를 줄이면 더 작은 음수가 나오면서 0에서 더 멀어진다. 따라서 합이 음수일 때는 **left** 인덱스를 올려서 더 높은 합을 만들어야 한다.

반대로 합이 양수인 상황에서 **left**를 증가 시키면 더 큰 양수가 나오면서 0에서 더 멀어진다. 따라서 합이 양수일 경우에는 **right**를 줄여서 더 작은 양수를 만들어야 한다.

예) loop 2: $\text{abs}(-2 + 98)$ 이므로 96이므로 0에서 더 가까워지기 위해 96대신 4를 더해야 하므로 **right**를 내려야 한다.

loop/ value	-99	-2	-1	4	98
loop1	left				right
loop2		left			right
loop3		left		right	
loop4		left	right		
loop5			left, right		



Hint

Step 1. 문제 유형



1. 투 포인터

기본 구조

left = 0

right = N - 1

while left < right:

total = arr[left] + arr[right]

if 조건 만족:

정답 갱신

break or 포인터 이동

elif total < 목표값:

left += 1

else:

right -= 1

✨ Solution



투 포인터 풀이

```
import sys

N = int(sys.stdin.readline())
arr = sorted(map(int, sys.stdin.readline().split()))

left = 0
right = N - 1

answer = (arr[left], arr[right], abs(arr[left] + arr[right]))

while left < right:
    total = arr[left] + arr[right]
    abs_total = abs(total)

    if abs_total < answer[2]:
        answer = (arr[left], arr[right], abs_total)
        if abs_total == 0:
            break

    if total > 0:
        right -= 1
    else:
        left += 1

print(f"{answer[0]} {answer[1]}")
```




Assignment

백준 #2473. 세 용액 (골드 3)

발제문제와 비슷한 유형인데 인덱스가 하나 더 늘어난 만큼 난이도가 올라간 문제