

[백준]2473_세 용액

▼ 상태	진행 중
📅 날짜	@2022년 4월 19일
≡ 공부유형	스터디
≡ 알고리즘	이진탐색
▼ 사이트	백준
☑ 깃허브	<input type="checkbox"/>

2473번: 세 용액

KOI 부설 과학연구소에서는 많은 종류의 산성 용액과 알칼리성 용액을 보유하고 있다. 각 용액에는 그 용액의 특성을 나타내는 하나의 정수가 주어져있다. 산성 용액의 특성값은 1부터 1,000,000,000까지의

<https://www.acmicpc.net/problem/2473>

BAE<JOON>
ONLINE JUDGE

개념

이진탐색

작업	메소드	ArrayList	LinkedList
add at last index	add()	O(1)	O(1)
add at given index	add(index, value)	O(N)	O(1)
remove by index	remove(index)	O(N)	O(1)
remove by value	remove(value)	O(N)	O(1)
get by index	get(index)	O(1)	O(N)
search by value	indexOf(value)	O(N)	O(N)

풀이

-10억~10억 사이의 정수가 (값이 3개가 최대로 크게 더하면 30억이라 int 안에서 해결 X -> long 사용)

3~5000 사이의 개수만큼 : 리스트 길이 = 이만큼 탐색 진행

Testcase 1. 답: -97 -2 98

```
5
-2 6 -97 -6 98
```

-97 -6 -2 6 98

- 생각 1. 용액 갯수만큼 for문을 돌려서 1개씩 그것과 이진탐색으로 2개를 고른 합의 최소를 고른다.
 - 투포인터 : $O(N)$
 - 투포인터를 갯수만큼 해줘야하므로 최대 $O(N^2)$
- 생각 2. 탐색을 두번한다.
 - 탐색으로 2개를 고른 뒤 그것의 더한 값을 배열에 넣어주고 →
 - 또 탐색으로 0에 가까운 것을 고른다.
 - 값을 넣어주고 정렬해주고 아니면 그값을 다시 빼야 함
 - 중간에서 빼는 것이 나은 LinkedList를 사용한다.?
 - or 계속 접근해서 값을 확인해야하므로 ArrayList를 사용한다?
 - 생각해보니까...앞의 것이 잘못된 선택일 경우 두번째 고른 값이 아닐수도 있을 것 같음...

Testcase 2. 답: -1000000000 -1000000000 -999999999

```
5
-999999999 -1000000000 -1000000000 -1000000000 -1000000000
```

- long형을 주의할 것!

코드

```
package BinarySearch;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.StringTokenizer;

// 투 포인터

public class BinarySearch_BOJ_2473_LJE {

    static void BinarySearch(int index) {
        int start = index+1; // 시작
        int end = N - 1; // 끝

        while (start < end) {
            long sum = (long)list[start] + (long)list[end] + (long)list[index];

            if (sum == 0) { // 0이면 바로 끝내기
                liquid[0] = list[index];
                liquid[1] = list[start]; //list.get(start);
                liquid[2] = list[end]; //list.get(end);
                break;
            }
            if (Math.abs(sum) < min) { // 새로운 합이 더 작다면 값 저장
                //무조건 index<start<end 순의 크기일 것이므로 차례대로 저장함으로써
                //출력 전 정렬을 줄여줄 수 있다.
                liquid[0] = list[index];
                liquid[1] = list[start]; //list.get(start);
                liquid[2] = list[end]; //list.get(end);
                min = Math.abs(sum);
            }

            if (sum >= 0) { // 합이 0보다 크다면 0기준 밸런스가 오른쪽에 치우친거니까
                end -= 1; // 오른쪽을 줄여주기
            } else { // 0보다 작으면 왼쪽에 치우친거니까
                start += 1; // 왼쪽을 줄여주기
            }
        }
    }

    static int N, liquid[], list[];
    static long min = Long.MAX_VALUE;
    // static List<Integer> list;
```

```

public static void main(String[] args) throws NumberFormatException, IOException {
    /* 입력 */
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    N = Integer.parseInt(br.readLine());
    list = new int[N]; //용액들 저장
    liquid = new int[3]; // 출력할 세 용액

    StringTokenizer st = new StringTokenizer(br.readLine());
    for (int i = 0; i < N; i++) {
        list[i] = Integer.parseInt(st.nextToken());
    }

    /* 탐색 */
    Arrays.sort(list); // 이진 탐색은 항상 정렬 후
    for (int i = 0; i < N; i++) { //for문을 돌며 값 하나씩 나머지에 대한 탐색하기
        BinarySearch(i);
    }

    // Arrays.sort(liquid); // 오름차순으로 출력 위해 정렬
    System.out.println(liquid[0] + " " + liquid[1] + " " + liquid[2]);

}
}

```