All Contests > e14csd > Closest Numbers

Closest Numbers

| Problem | Submissions | Leaderboard |

Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well. In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

For example, if you've got the list $[5, 2, 3, 4, 1]$, sort it as $[1, 2, 3, 4, 5]$ to see that several pairs have the minimum difference of 1: $[(1, 2), (2, 3), (3, 4), (4, 5)]$. The return array would be $[1, 2, 2, 3, 3, 4, 4, 5]$.

Given a list of unsorted integers, arr, find the pair of elements that have the smallest absolute difference between them. If there are multiple pairs, find them all.

Function Description

Complete the *closestNumbers* function in the editor below. It must return an array of integers as described.

closestNumbers has the following parameter(s):

- *arr*: an array of integers

Input Format

The first line contains a single integer n, the length of arr.
The second line contains n space-separated integers, $arr[i]$.

Constraints

- $2 \leq n \leq 200000$

- $-10^7 \leq arr[i] \leq 10^7$

- All $a[i]$ are unique in arr.

Output Format

Output the pairs of elements with the smallest difference. If there are multiple pairs, output all of them in ascending order, all on the same line with just a single space between each pair of numbers. A number may be part of two pairs when paired with its predecessor and its successor.

Sample Input 0

```
10
-20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594 266854
```

Sample Output 0

```
-20 30
```

Explanation 0

(30) - (-20) = 50, which is the smallest difference.

Sample Input 1

```
12
-20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594 266854 -520 -470
```

Sample Output 1

```
-520 -470 -20 30
```

Explanation 1

(-470) - (-520) = 30 - (-20) = 50, which is the smallest difference.

Sample Input 2

```
4
5 4 3 2
```

Sample Output 2

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

Explanation 2

Here, the minimum difference will be *1*. Valid pairs are *(2, 3)*, *(3, 4)*, and *(4, 5)*. We print the elements of each pair, space-separated on a single line.

f   y   in

Contest ends in 13 minutes

Submissions: 2
Max Score: 35

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

Python 3

```python
#!/bin/python3

import math
import os
import random
import re
import sys

# Complete the closestNumbers function below.
def closestNumbers(arr):
    s = sorted(arr)
    d = []
    for i in range(len(s) - 1):
        d.append(s[i+1] - s[i])

    v = min(d)

    ans = []

    for i in range(len(s) - 1):
        if (s[i+1] - s[i]) == v:
            ans.append(s[i])
            ans.append(s[i+1])

    return ans

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
```

```
29
30      n = int(input())
31
32      arr = list(map(int, input().rstrip().split()))
33
34      result = closestNumbers(arr)
35
36      fptr.write(' '.join(map(str, result)))
37      fptr.write('\n')
38
39      fptr.close()
40
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Testcase 0 ✔	Testcase 1 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
10
-20 -3916237 -357920 -3620601 7374819 -7330761 30 6246457 -6461594 266854
```

Your Output (stdout)

```
-20 30
```

Expected Output

```
-20 30
```