PRACTICE COMPETE JOBS LEADERBOARD Search pruthvishalcodi1

All Contests > ALCoding Summer Weekly Contest 3 > Find the Running Median

Find the Running Median

locked

by amititkgp

Problem Submissions Leaderboard Discussions Editorial

The median of a set of integers is the midpoint value of the data set for which an equal number of integers are less than and greater than the value. To find the median, you must first sort your set of integers in non-decreasing order, then:

- If your set contains an odd number of elements, the median is the middle element of the sorted sample. In the sorted set $\{1, 2, 3\}$, 2 is the median.
- If your set contains an even number of elements, the median is the average of the two middle elements of the sorted sample. In the sorted set $\{1, 2, 3, 4\}$, $\frac{2+3}{2} = 2.5$ is the median.

Given an input stream of n integers, you must perform the following task for each i^{th} integer:

1. Add the i^{th} integer to a running list of integers.
2. Find the median of the updated list (i.e., for the first element through the i^{th} element).
3. Print the list's updated median on a new line. The printed value must be a double-precision number scaled to 1 decimal place (i.e., 12.3 format).

Input Format

The first line contains a single integer, n, denoting the number of integers in the data stream.
Each line i of the n subsequent lines contains an integer, a_i, to be added to your list.

Constraints

- $1 \leq n \leq 10^5$
- $0 \leq a_i \leq 10^5$

Output Format

After each new integer is added to the list, print the list's updated median on a new line as a single double-precision number scaled to 1 decimal place (i.e., 12.3 format).

Sample Input

```
6
12
4
5
3
8
7
```

Sample Output

```
12.0
8.0
5.0
4.5
5.0
6.0
```

Explanation

There are $n = 6$ integers, so we must print the new median on a new line as each integer is added to the list:

1. $list = \{12\}, median = 12.0$
2. $list = \{12, 4\} \rightarrow \{4, 12\}, median = \frac{12+4}{2} = 8.0$
3. $list = \{12, 4, 5\} \rightarrow \{4, 5, 12\}, median = 5.0$
4. $list = \{12, 4, 5, 3\} \rightarrow \{3, 4, 5, 12\}, median = \frac{4+5}{2} = 4.5$
5. $list = \{12, 4, 5, 3, 8\} \rightarrow \{3, 4, 5, 8, 12\}, median = 5.0$
6. $list = \{12, 4, 5, 3, 8, 7\} \rightarrow \{3, 4, 5, 7, 8, 12\}, median = \frac{5+7}{2} = 6.0$

Submissions: 62
Max Score: 25

Rate This Challenge:
☆☆☆☆☆

More

Current Buffer (saved locally, editable)

C

```
#include <assert.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char* readline();

/*
 * Complete the runningMedian function below.
 */

/*
 * Please store the size of the double array to be returned in result_count pointer. For example,
 * double a[3] = {1.1, 2.2, 3.3};
 *
 * *result_count = 3;
 *
 * return a;
 *
 */
double* runningMedian(int a_count, int* a, int* result_count) {
    /*
     * Write your code here.
     */

}

int main()
{
    FILE* fptr = fopen(getenv("OUTPUT_PATH"), "w");

    char* a_count_endptr;
    char* a_count_str = readline();
    int a_count = strtol(a_count_str, &a_count_endptr, 10);

    if (a_count_endptr == a_count_str || *a_count_endptr != '\0') { exit(EXIT_FAILURE); }

    int a[a_count];

    for (int a_itr = 0; a_itr < a_count; a_itr++) {
        char* a_item_endptr;
        char* a_item_str = readline();
        int a_item = strtol(a_item_str, &a_item_endptr, 10);

        if (a_item_endptr == a_item_str || *a_item_endptr != '\0') { exit(EXIT_FAILURE); }

        a[a_itr] = a_item;
    }

    int result_count;
    double* result = runningMedian(a_count, a, &result_count);

    for (int result_itr = 0; result_itr < result_count; result_itr++) {
        fprintf(fptr, "%lf", result[result_itr]);

        if (result_itr != result_count - 1) {
            fprintf(fptr, "\n");
        }
    }

    fprintf(fptr, "\n");

    fclose(fptr);

    return 0;
}

char* readline() {
    size_t alloc_length = 1024;
    size_t data_length = 0;
    char* data = malloc(alloc_length);

    while (true) {
        char* cursor = data + data_length;
        char* line = fgets(cursor, alloc_length - data_length, stdin);

        if (!line) { break; }

        data_length += strlen(cursor);

        if (data_length < alloc_length - 1 || data[data_length - 1] == '\n') { break; }

        size_t new_length = alloc_length << 1;
        data = realloc(data, new_length);

        if (!data) { break; }

        alloc_length = new_length;
    }

    if (data[data_length - 1] == '\n') {
        data[data_length - 1] = '\0';
    }

    data = realloc(data, data_length);

    return data;
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature