Your Almost Sorted submission got 50.00 points. Share Tweet

Try the Next Challenge | Try a Random Challenge

Almost Sorted

by PRASHANTB1984

Problem Submissions Leaderboard Discussions Editorial

Given an array with n elements, can you sort this array in *ascending order* using only one of the following operations?

1. Swap two elements.
2. Reverse one sub-segment.

Input Format

The first line contains a single integer, n, which indicates the size of the array.
The next line contains n integers separated by spaces.

```
n
d1 d2 ... dn
```

Constraints

$2 \leq n \leq 100000$
$0 \leq d_i \leq 1000000$
All d_i are distinct.

Output Format

1. If the array is already sorted, output *yes* on the first line. You do not need to output anything else.

1. If you can sort this array using one single operation (from the two permitted operations) then output *yes* on the first line and then:

   a. If you can sort the array by swapping d_l and d_r, output *swap l r* in the second line. l and r are the indices of the elements to be swapped, assuming that the array is indexed from 1 to n.

   b. Else if it is possible to sort the array by reversing the segment $d[l \ldots r]$, output *reverse l r* in the second line. l and r are the indices of the first and last elements of the subsequence to be reversed, assuming that the array is indexed from 1 to n.

   $d[l \ldots r]$ represents the sub-sequence of the array, beginning at index l and ending at index r, both inclusive.

   If an array can be sorted by either swapping or reversing, stick to the swap-based method.

2. If you cannot sort the array in either of the above ways, output *no* in the first line.

Sample Input #1

```
2
4 2
```

Sample Output #1

```
yes
swap 1 2
```

Sample Input #2

```
3
3 1 2
```

Sample Output #2

```
no
```

Sample Input #3

```
6
1 5 4 3 2 6
```

Sample Output #3

```
yes
reverse 2 5
```

Explanation

For #1, you can both *swap(1, 2)* and *reverse(1, 2)*, but if you can sort the array using swap, output swap only.
For #2, it is impossible to sort by one single operation (among those permitted).
For #3, you can reverse the sub-array *d[2...5] = "5 4 3 2"*, then the array becomes sorted.

Medium Submitted 16563 times
Max Score **50**

Need Help?

View Discussions

View Editorial Solution

View Top Submissions

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) C++14

```
#include <iostream>
#include <vector>
#include <algorithm>

int main()
{
    int Size; std::cin >> Size;
    std::vector<int> vec(Size);
    for(auto &it: vec) std::cin >> it;

    if(std::is_sorted(vec.begin(), vec.end()))
        std::cout << "yes" << std::endl;
    else
    {
```

```cpp
        std::vector<std::pair<int,int>> Front;
        std::vector<std::pair<int,int>> Back;

        int rev = Size - 1;
        for(int i = 0; i < Size-1; ++i)
        {
            if(vec[i] > vec[i+1])
                Front.emplace_back(std::make_pair(vec[i], i+1));

            if(vec[rev] < vec[rev-1])
                Back.emplace_back(std::make_pair(vec[rev], rev+1));

            --rev;
        }
        if(Front.size() <= 2 && Back.size() <= 2)
        {
            std::swap(vec[Front[0].second-1],vec[Back[0].second-1]);
            if(std::is_sorted(vec.begin(), vec.end()))
            {
                std::cout << "yes\n" ;
                std::cout << "swap " << Front[0].second << " " << Back[0].second <<"\n";
            }
            else std::cout << "no" << std::endl;
        }
        else
        {
            Front.emplace_back(std::make_pair(Back[0].first, Back[0].second));
            if(std::is_sorted(Front.rbegin(), Front.rend()))
            {
                //if( (Front.size() == vec.size())
                std::cout << "yes\n" ;
                std::cout << "reverse " << Front[0].second << " " << Back[0].second <<"\n";
            }
            else std::cout << "no" << std::endl;
        }
        /*for(const auto &it: Front) std::cout<<it.first<<" "<<it.second<<"\n";
        std::cout<<std::endl;
        for(const auto &it: Back) std::cout<<it.first<<" "<<it.second<<"\n";*/
    }
    return 0;
}

```

Line: 56 Col: 1

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

Congrats, you solved this challenge!

Challenge your friends:

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5
✔ Test Case #6	✔ Test Case #7	✔ Test Case #8
✔ Test Case #9	✔ Test Case #10	✔ Test Case #11
✔ Test Case #12	✔ Test Case #13	✔ Test Case #14
✔ Test Case #15	✔ Test Case #16	✔ Test Case #17
✔ Test Case #18	✔ Test Case #19	✔ Test Case #20
✔ Test Case #21	✔ Test Case #22	✔ Test Case #23

✔ Test Case #24

You've earned 50.00 points. Next Challenge