Problem: Almost Sorted

Given an array with  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, , which indicates the size of the array.   
The next line contains  integers separated by spaces.

n

d1 d2 ... dn

**Constraints**   
   
    
All  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  and , output *swap l r* in the second line.  and  are the indices of the elements to be swapped, assuming that the array is indexed from  to .

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

 represents the sub-sequence of the array, beginning at index  and ending at index , both inclusive.

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

1. 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.

Solution

int sort(long array[], long length)

{

for(long i=0; i<length; i++)

{if(array[i]>array[i+1])

{return 0;

break;}

}

return 1;

}

int main()

{

long length, start, end, bound, flag,count=0;

cin>>length;

long array[length];

for(int i=0; i<length; i++)

{cin>>array[i];}

if(sort(array,length)==1)

{cout<<"yes"; return 0;}

else

{

//Finding the endpoints

long i=0;

flag=0;

while(flag!=1 && i<length)

{

if(array[i]>array[i+1])

{flag=1; start=i; }

i++;

}

flag=0;

i=length-1;

while(flag!=1 && i>=0)

{

if(array[i]<array[i-1])

{flag=1;

end=i;

}

i--;

}

bound=(end-start+1)/2;

for(i=0; i<bound; i++)

{ if(array[start+i]>array[start+i+1] && array[end-i]<array[end-i-1])

{ long temp=array[end-i];

array[end-i]=array[start+i];

array[start+i]=temp;

count++;

}

}

if(sort(array,length)==1)

{cout<<"yes"<<endl;

if(count==1)

{cout<<"swap "<<start+1<<" "<<end+1;}

else if(count>0)

{cout<<"reverse "<<start+1<<" "<<end+1;}

}

else

{cout<<"no"; }

}

return 0;

}

`’Anshul AgGarwal