**Министерство образования и науки Российской Федерации**
**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Дисциплина «**Алгоритмы и структуры данных**»

# ОТЧЁТ

по лабораторным работам за 2 неделю

Студент **Пастухов К.А. группы P3218**

**Санкт-Петербург**
**2018 г.**

# Сортировка слиянием

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью сортировки слиянием.

## Выполнение

```
def read_file():
    with open('input.txt', "r") as file:
        size = int(file.readline())
        data = [int(num) for num in file.readline().split(' ')]
    return size, data


def merge(left, right, left_border, right_border, file):
    result = []
    i, j = 0, 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result.extend(left[i:])
    result.extend(right[j:])
    file.write(f"{left_border + 1} {right_border + 1} {result[0]} {result[-1]}"
+ '\n')
    return result


def merge_sort(data, left_border, right_border, file):
    if len(data) == 1:
        return data
    middle = len(data) // 2
    left_data = merge_sort(data[:middle], left_border, left_border + middle - 1,
file)
    right_data = merge_sort(data[middle:], left_border + middle, right_border,
file)
    return merge(left_data, right_data, left_border, right_border, file)


def main():
    size, data = read_file()
    file = open('output.txt', 'w')
    file.write(" ".join(str(r) for r in merge_sort(data, 0, size - 1, file)))


if __name__ == '__main__':
    main()
```

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 0.718 | 25088000 | 1039245 | 4403712 |
| 1 | OK | 0.031 | 9347072 | 25 | 104 |
| 2 | OK | 0.031 | 9383936 | 6 | 1 |
| 3 | OK | 0.046 | 9314304 | 8 | 12 |
| 4 | OK | 0.031 | 9371648 | 8 | 12 |
| 5 | OK | 0.031 | 9338880 | 42 | 153 |
| 6 | OK | 0.046 | 9322496 | 43 | 153 |
| 7 | OK | 0.031 | 9330688 | 51 | 177 |
| 8 | OK | 0.031 | 9330688 | 45 | 160 |
| 9 | OK | 0.031 | 9338880 | 105 | 329 |
| 10 | OK | 0.031 | 9232384 | 110 | 342 |
| 11 | OK | 0.046 | 9420800 | 107 | 335 |
| 12 | OK | 0.031 | 9453568 | 461 | 2043 |
| 13 | OK | 0.046 | 9416704 | 560 | 2330 |
| 14 | OK | 0.046 | 9424896 | 388 | 1821 |
| 15 | OK | 0.031 | 9433088 | 408 | 1882 |
| 16 | OK | 0.031 | 9375744 | 1042 | 3775 |
| 17 | OK | 0.031 | 9428992 | 1043 | 3783 |
| 18 | OK | 0.046 | 9453568 | 1044 | 3774 |
| 19 | OK | 0.046 | 9564160 | 5587 | 25512 |
| 20 | OK | 0.046 | 9531392 | 6733 | 28936 |
| 21 | OK | 0.031 | 9580544 | 4737 | 22959 |
| 22 | OK | 0.031 | 9502720 | 5685 | 25798 |
| 23 | OK | 0.046 | 9568256 | 10383 | 39967 |
| 24 | OK | 0.046 | 9551872 | 10421 | 40059 |
| 25 | OK | 0.046 | 9580544 | 10420 | 40056 |
| 26 | OK | 0.093 | 11034624 | 65880 | 305387 |
| 27 | OK | 0.109 | 10915840 | 77550 | 340375 |
| 28 | OK | 0.093 | 10997760 | 57488 | 280212 |
| 29 | OK | 0.140 | 10915840 | 68090 | 311996 |
| 30 | OK | 0.093 | 10956800 | 103872 | 420188 |
| 31 | OK | 0.093 | 10960896 | 103940 | 420365 |
| 32 | OK | 0.093 | 11026432 | 103842 | 420121 |
| 33 | OK | 0.562 | 24178688 | 758839 | 3554250 |
| 34 | OK | 0.562 | 24576000 | 875802 | 3905101 |
| 35 | OK | 0.578 | 25088000 | 675241 | 3303453 |
| 36 | OK | 0.578 | 23629824 | 782803 | 3626112 |
| 37 | OK | 0.703 | 24293376 | 1038992 | 4403247 |
| 38 | OK | 0.703 | 24354816 | 1038702 | 4402562 |
| 39 | OK | 0.718 | 23707648 | 1039245 | 4403712 |

## Число инверсий

Дан массив целых чисел. Ваша задача — подсчитать число инверсий в нем.

Подсказка: чтобы сделать это быстрее, можно воспользоваться модификацией сортировки слиянием.

## Выполнение

```
def merge_sort(data):
    if len(data) > 1:
        middle = len(data) // 2
        left = data[:middle]
        right = data[middle:]
        inversion = merge_sort(left) + merge_sort(right)
```

```
        l, r = 0, 0
        cursor = 0
        while l < len(left) and r < len(right):
            if left[l] <= right[r]:
                data[cursor] = left[l]
                l += 1
            else:
                inversion += len(left) - l
                data[cursor] = right[r]
                r += 1
            cursor += 1
        while l < len(left):
            data[cursor] = left[l]
            cursor += 1
            l += 1
        while r < len(right):
            data[cursor] = right[r]
            cursor += 1
            r += 1
        return inversion
    return 0


def read_file():
    with open("input.txt", 'r')as file:
        size = int(file.readline())
        data = [int(num) for num in file.readline().split(" ")]
    return size, data


def main():
    size, data = read_file()
    res = open("output.txt", "w")
    res.write(str(merge_sort(data)))


if __name__ == '__main__':
    main()
```

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 0.531 | 21782528 | 1039245 | 10 |
| 1 | OK | 0.031 | 8744960 | 25 | 2 |
| 2 | OK | 0.031 | 8728576 | 6 | 1 |
| 3 | OK | 0.031 | 8761344 | 8 | 1 |
| 4 | OK | 0.031 | 8822784 | 8 | 1 |
| 5 | OK | 0.031 | 8781824 | 42 | 1 |
| 6 | OK | 0.062 | 8708096 | 43 | 2 |
| 7 | OK | 0.031 | 8740864 | 51 | 1 |
| 8 | OK | 0.031 | 8790016 | 45 | 2 |
| 9 | OK | 0.031 | 8761344 | 105 | 2 |
| 10 | OK | 0.031 | 8781824 | 110 | 2 |
| 11 | OK | 0.046 | 8675328 | 107 | 2 |
| 12 | OK | 0.046 | 8712192 | 461 | 1 |
| 13 | OK | 0.031 | 8732672 | 560 | 4 |
| 14 | OK | 0.031 | 8699904 | 388 | 1 |
| 15 | OK | 0.062 | 8744960 | 408 | 4 |
| 16 | OK | 0.031 | 8777728 | 1042 | 4 |
| 17 | OK | 0.062 | 8769536 | 1043 | 4 |
| 18 | OK | 0.031 | 8728576 | 1044 | 4 |
| 19 | OK | 0.031 | 8830976 | 5587 | 1 |
| 20 | OK | 0.031 | 8790016 | 6733 | 6 |
| 21 | OK | 0.062 | 8867840 | 4737 | 1 |
| 22 | OK | 0.046 | 8794112 | 5685 | 6 |
| 23 | OK | 0.031 | 8802304 | 10383 | 6 |
| 24 | OK | 0.062 | 8830976 | 10421 | 6 |
| 25 | OK | 0.031 | 8855552 | 10420 | 6 |
| 26 | OK | 0.062 | 9605120 | 65880 | 1 |
| 27 | OK | 0.062 | 9650176 | 77550 | 8 |
| 28 | OK | 0.062 | 9650176 | 57488 | 1 |
| 29 | OK | 0.109 | 9682944 | 68090 | 8 |
| 30 | OK | 0.062 | 9744384 | 103872 | 8 |
| 31 | OK | 0.125 | 9715712 | 103940 | 8 |
| 32 | OK | 0.062 | 9715712 | 103842 | 8 |
| 33 | OK | 0.375 | 21192704 | 758839 | 1 |
| 34 | OK | 0.453 | 21663744 | 875802 | 10 |
| 35 | OK | 0.375 | 21151744 | 675241 | 1 |
| 36 | OK | 0.453 | 21180416 | 782803 | 10 |
| 37 | OK | 0.531 | 21626880 | 1038992 | 10 |
| 38 | OK | 0.515 | 21782528 | 1038702 | 10 |
| 39 | OK | 0.515 | 21630976 | 1039245 | 10 |

## Анти-quick sort

Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число таких сравнений.

## Выполнение

```
def read_file():
    with open('input.txt', 'r') as file:
        size = int(file.readline())
    return size


def main():
    size = read_file()
    numbers = [i for i in range(1, size + 1)]
    for i in range(size):
        if (i > 1):
            numbers[i], numbers[i // 2] = numbers[i // 2], numbers[i]
    return numbers
```

```
def write_file(res):
    with open('output.txt', 'w') as file:
        file.write(' '.join(str(r) for r in res))


if __name__ == '__main__':
    write_file(main())
```

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 1.109 | 123154432 | 9 | 6888895 |
| 1 | OK | 0.031 | 9302016 | 3 | 5 |
| 2 | OK | 0.046 | 9224192 | 3 | 1 |
| 3 | OK | 0.078 | 9236480 | 3 | 3 |
| 4 | OK | 0.031 | 9367552 | 3 | 7 |
| 5 | OK | 0.031 | 9302016 | 3 | 9 |
| 6 | OK | 0.046 | 9211904 | 3 | 11 |
| 7 | OK | 0.046 | 9273344 | 3 | 13 |
| 8 | OK | 0.046 | 9256960 | 3 | 15 |
| 9 | OK | 0.046 | 9273344 | 3 | 17 |
| 10 | OK | 0.015 | 9281536 | 4 | 20 |
| 11 | OK | 0.046 | 9310208 | 4 | 35 |
| 12 | OK | 0.046 | 9302016 | 5 | 291 |
| 13 | OK | 0.093 | 9334784 | 6 | 3892 |
| 14 | OK | 0.062 | 10522624 | 7 | 48899 |
| 15 | OK | 0.046 | 10424320 | 7 | 48893 |
| 16 | OK | 0.156 | 24555520 | 8 | 756194 |
| 17 | OK | 0.281 | 36900864 | 8 | 1556238 |
| 18 | OK | 0.515 | 62947328 | 8 | 3151811 |
| 19 | OK | 1.109 | 123154432 | 8 | 6888887 |
| 20 | OK | 1.093 | 123109376 | 9 | 6888895 |

## K-ая порядковая статистика

Дан массив из *n* элементов. Какие числа являются *k1* (*k1* + 1),..., *k2 в порядке неубывания в этом массиве?*

## Выполнение

```
using System.IO;

namespace week2_lab4
{
    internal static class Program
    {
        private static void quick_sort(int[] data, int left, int right, int k1,
int k2)
```

```
{
    while (true)
    {
        if (left > k2 || right < k1) return;
        int i = left, j = right;
        var x = data[(left + right) / 2];

        while (i <= j)
        {
            while (data[i].CompareTo(x) < 0)
            {
                i++;
            }

            while (data[j].CompareTo(x) > 0)
            {
                j--;
            }

            if (i > j) continue;
            var tmp = data[i];
            data[i] = data[j];
            data[j] = tmp;

            i++;
            j--;
        }

        if (left < j)
        {
            quick_sort(data, left, j, k1, k2);
        }

        if (i < right)
        {
            left = i;
            continue;
        }

        break;
    }
}


private static void Main()
{
    int n, k1, k2;
    int a, b, c;
    int[] data;
    using (var reader = new StreamReader("input.txt"))
    {
        var firstLine = reader.ReadLine()?.Split();
        n = int.Parse(firstLine?[0]);
        k1 = int.Parse(firstLine?[1]) - 1;
        k2 = int.Parse(firstLine?[2]) - 1;
        var secondLine = reader.ReadLine()?.Split();
        a = int.Parse(secondLine?[0]);
        b = int.Parse(secondLine?[1]);
        c = int.Parse(secondLine?[2]);
        data = new int[n];
```

```
                data[0] = (int.Parse(secondLine?[3]));
                data[1] = (int.Parse(secondLine?[4]));
            }
            for (var i = 2; i < n; i++)
            {
                data[i] = (a * data[i - 2] + b * data[i - 1] + c);
            }


            using (var writer = new StreamWriter("output.txt"))
            {
                quick_sort(data, 0, n - 1, k1, k2);
                for (var i = k1; i <= k2; i++)
                {
                    writer.Write($"{data[i]} ");
                }
            }
        }
    }
}
```

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 0.812 | 170790912 | 54 | 2400 |
| 1 | OK | 0.031 | 10383360 | 18 | 6 |
| 2 | OK | 0.031 | 10379264 | 28 | 9 |
| 3 | OK | 0.031 | 10477568 | 32 | 4 |
| 4 | OK | 0.015 | 10366976 | 33 | 5 |
| 5 | OK | 0.031 | 10412032 | 32 | 10 |
| 6 | OK | 0.031 | 10383360 | 33 | 5 |
| 7 | OK | 0.031 | 10424320 | 32 | 19 |
| 8 | OK | 0.031 | 10391552 | 32 | 21 |
| 9 | OK | 0.031 | 10424320 | 25 | 300 |
| 10 | OK | 0.031 | 10436608 | 22 | 382 |
| 11 | OK | 0.031 | 10412032 | 23 | 477 |
| 12 | OK | 0.031 | 10391552 | 35 | 12 |
| 13 | OK | 0.015 | 10444800 | 38 | 11 |
| 14 | OK | 0.046 | 10452992 | 36 | 1074 |
| 15 | OK | 0.015 | 10412032 | 36 | 561 |

## Сортировка пугалом

«Сортировка пугалом» — это давно забытая народная потешка, которую восстановили по летописям специалисты платформы «Открытое образование» специально для этого курса.

## Выполнение

```
def quick_sort(nums, fst, lst, k):
```

```
    if fst >= lst:
        return

    i, j = fst, lst
    x = nums[(lst - fst) // (k * 2) * k + fst]
   while i <= j:
        while nums[i] < x:
            i += k
        while nums[j] > x:
            j -= k
        if i <= j:
            nums[i], nums[j] = nums[j], nums[i]
            i, j = i + k, j - k
    quick_sort(nums, fst, j, k)
    quick_sort(nums, i, lst, k)


def read_file():
    with open('input.txt', "r") as file:
        size = [int(s) for s in file.readline().split(' ')]
        data = [int(num) for num in file.readline().split(' ')]
    return size, data


size, data = read_file()
n = size[0]
k = size[1]

offset = n - n % k
for i in range(k):
    index = i + offset
    quick_sort(data, i, index if index < n else index - k, k)

result = True

for i in range(n-1):
    if data[i] > data[i+1]:
        result = False
        break
with open("output.txt", "w") as res:
    res.write("YES" if result else "NO")
```

| № теста | Результат | Время, с | Память | Размер входного файла | Размер выходного файла |
|---|---|---|---|---|---|
| Max | | 0.375 | 21884928 | 1039313 | 3 |
| 1 | OK | 0.031 | 8736768 | 12 | 2 |
| 2 | OK | 0.031 | 8724480 | 16 | 3 |
| 3 | OK | 0.031 | 8794112 | 112 | 3 |
| 4 | OK | 0.031 | 8736768 | 111 | 2 |
| 5 | OK | 0.031 | 8708096 | 112 | 3 |
| 6 | OK | 0.046 | 8749056 | 112 | 2 |
| 7 | OK | 0.031 | 8720384 | 109 | 3 |
| 8 | OK | 0.031 | 8753152 | 112 | 2 |
| 9 | OK | 0.031 | 8818688 | 110 | 3 |
| 10 | OK | 0.031 | 8572928 | 111 | 2 |
| 11 | OK | 0.062 | 8769536 | 108 | 3 |
| 12 | OK | 0.031 | 8839168 | 11674 | 3 |
| 13 | OK | 0.046 | 8757248 | 11707 | 2 |
| 14 | OK | 0.031 | 8826880 | 11712 | 3 |
| 15 | OK | 0.031 | 8806400 | 11754 | 2 |
| 16 | OK | 0.031 | 8871936 | 11708 | 3 |
| 17 | OK | 0.031 | 8818688 | 11740 | 2 |
| 18 | OK | 0.031 | 8859648 | 11726 | 3 |
| 19 | OK | 0.046 | 8855552 | 11680 | 2 |
| 20 | OK | 0.031 | 8802304 | 11741 | 3 |
| 21 | OK | 0.078 | 10072064 | 128736 | 3 |
| 22 | OK | 0.062 | 9990144 | 128832 | 2 |
| 23 | OK | 0.062 | 10043392 | 128751 | 3 |
| 24 | OK | 0.031 | 10039296 | 128866 | 2 |
| 25 | OK | 0.046 | 10018816 | 128700 | 3 |
| 26 | OK | 0.062 | 10129408 | 128707 | 2 |
| 27 | OK | 0.062 | 10113024 | 128729 | 3 |
| 28 | OK | 0.046 | 10047488 | 128807 | 2 |
| 29 | OK | 0.046 | 10059776 | 128784 | 3 |
| 30 | OK | 0.375 | 20467712 | 1039313 | 3 |
| 31 | OK | 0.359 | 20508672 | 1038610 | 2 |
| 32 | OK | 0.375 | 21884928 | 1038875 | 3 |
| 33 | OK | 0.109 | 21610496 | 1038723 | 2 |
| 34 | OK | 0.125 | 21610496 | 1038749 | 3 |
| 35 | OK | 0.328 | 20131840 | 1038747 | 2 |
| 36 | OK | 0.375 | 21860352 | 1039043 | 3 |
| 37 | OK | 0.140 | 21839872 | 1039210 | 2 |
| 38 | OK | 0.156 | 21622784 | 1038967 | 3 |