

Федеральное государственное автономное образовательное
учреждение высшего образования
Университет ИТМО

Факультет инфокоммуникационных технологий

Алгоритмы и структуры данных:

**Отчёт по лабораторной работе №3: Быстрая
сортировка, сортировка за линейное время**

Выполнил:
Бочкарь Артём Артёмович

Группа: **K32392**

Преподаватели:
Артамонова В. Е.

Санкт-Петербург 2023 г.

Задача №1: Улучшение Quick Sort

В первом задании нужно было написать улучшенный Quick Sort, который работает эффективно даже с последовательностями, содержащими большое количество одинаковых элементов:

```
import random
import time
def partition3(G, l, r):
    x = G[l]
    j=1
    k=1
    for i in range(l + 1, r):
        if G[i] < x:
            j += 1
            k += 1
            G[j], G[i] = G[i], G[j]
        elif G[i] == x:
            k += 1
            G[k], G[i] = G[i], G[k]
            G[j], G[l] = G[l], G[j]
    return j, k
def randqs3(G, l, r):
    if l < r:
        k = random.randint(l, r-1)
        G[l], G[k] = G[k], G[l]
        m1, m2 = partition3(G, l, r)
        randqs3(G, l, m1)
        randqs3(G, m2 + 1, r)
    return G
n = int(input())
mass = list(map(int, input().split()))
start = time.perf_counter()
print(*randqs3(mass, 0, n))
print('Time spent: %s seconds' % (time.perf_counter() - start))
```

Тесты:

```
5
2 3 9 2 2
2 2 2 3 9
Time spent: 0.000108300000000083828 seconds
```

Задача №2: Anti-Quick Sort

В этом задании нужно было написать программу, которая генерирует тест, на котором быстрая сортировка сделает наибольшее число сравнений элементов:

```
import math
k = int(input())
a = [0] * k
for i in range(k):
    a[i] = i + 1
for i in range(2, k):
    a[i], a[math.ceil(i/2)] = a[math.ceil(i/2)], a[i]
for i in range(k):
    print(a[i], end=' ')
```

Тесты:



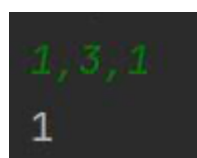
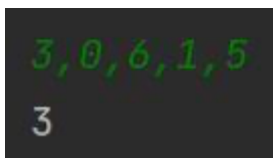
Задача №5: Индекс Хирша

В пятом задании нужно написать алгоритм для вычисления индекса Хирша:

```
import time
def hirsh(citat): s=0
    siz = len(citat)
    mp = [0]*(siz + 1)
    for i in citat:
        mp[min(i, siz)] += 1
    for j in range(siz, 0, -1):
        s += mp[j]
    if s >= j:
        return j

citat = list(map(int, input().split(',')))
start = time.perf_counter() print(hirsh(citat))
print('Time:', time.perf_counter() - start)
```

Тесты:



Задача №3: Сортировка пугалом

В третьем задании нужно было написать алгоритм сортировки пугалом. «Сортировка пугалом» - это давно забытая народная потешка. Участнику под верхнюю одежду продевают деревянную палку, так, что у него оказываются растопырены руки, как у огородного пугало. Перед ним ставится n матрёшек в ряд. Из-за палки единственное, что он может сделать – это взять в руки две матрёшки на расстоянии k друг от друга (то есть i -ую и $i + k$ -ую), развернуться и поставить их обратно в ряд, таким образом поменяв их местами. Если массив может быть оstarтирован таким способом, то выводится «ДА», а не если не может, то выводится «НЕТ»:

```
import time
def checkorder(n, g):
    for i in range(1, g):
        if n[i] < n[i - 1]:
            return 1
    return 0

def part(G, l, r, k):
    x = G[l]
    li = l
    ri = r

    while li <= ri:
        while n[li] < x:
            li += k
        while n[ri] > x:
            ri -= k

        if li <= ri:
            n[li], n[ri] = n[ri], n[li]
            li += k
            ri -= k

    return li

def quicksort(G, l, r, k):
    if l < r:
        m = part(G, l, r, k)
        quicksort(G, l, m - k, k)
        quicksort(G, m, r, k)

def pugalo(n, g, k):
    r = ((g - 1) // k) * k
    for i in range(k):
        quicksort(n, i, r, k)

if r > g - 1:
    r -= k

    return n
g, k = map(int, input().split())
n = list(map(int, input().split()))
pugalo(n, g, k)
start = time.perf_counter()
if checkorder(n, g):
    print("НЕТ")
else:
    print("ДА")
print('Time:', time.perf_counter() - start)
```

Тесты:

```
3 2
2 1 3
НЕТ
```

```
5 3
1 5 3 4 1
ДА
```

Задача №8: К ближайших точек к началу координат

В этом задании надо найти k ближайших точек к началу координат среди данных n точек.

```
import random
import time
n, k = list(map(int, input().split()))
fin = []
for i in range(n):
    x, y = list(map(int, input().split()))
    fin.append([x, y, ((x ** 2 + y ** 2) ** 0.5)])
start = time.perf_counter()
def part3(g, l, r):
    x = g[l][2]
    p=r
    j=l
    i=j
    while i <= p:
        if g[i][2] < x:
            g[j], g[i] = g[i], g[j]
            j += 1
        elif g[i][2] > x:
            g[p], g[i] = g[i], g[p]
            p -= 1
            i -= 1
        i += 1
    return j, p

def randqs3(A, l, r):
    if l < r:
        k = random.randint(l, r-1)
        A[l], A[k] = A[k], A[l]
        m1, m2 = part3(A, l, r)
        randqs3(A, l, m1)
        randqs3(A, m2 + 1, r)
    return A

fin = randqs3(fin, 0, len(fin)-1)
for t in range(k):
    print([fin[t][0], fin[t][1]], end='')
print("\nTime:", time.perf_counter() - start)
```

Тесты:

```
2 1
1 3
-2 2
[-2, 2]
```

```
3 2
3 3
5 -2
-2 4
[3, 3][-2, 4]
```