Министерство высшего образования и науки Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Отчет

По лабораторной работе №2

По курсу «Логика и основы алгоритмизации в инженерных задачах»

На тему «Оценка времени выполнения программ»

Работу выполнила студентка группы 20ВВ2:

Войнова Д.А.

Приняли:

Юрова О.В.

Митрохин М.А.

Пенза 2021

Листинг

#define\_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>;

#include<time.h>;

voidqs(int\* items, intleft, intright) //вызовфункции: qs(items, 0, count-1);

{

inti, j;

int x, y;

i = left; j = right;

/\* выбор компаранда \*/

x = items[(left + right) / 2];

do {

while ((items[i] < x) && (i<right)) i++;

while ((x <items[j]) && (j >left)) j--;

if (i<= j) {

y = items[i];

items[i] = items[j];

items[j] = y;

i++; j--;

}

} while (i<= j);

if (left< j) qs(items, left, j);

if (i<right) qs(items, i, right);

}

voidshell\_sort(int\* array, intsize) {

for (int s = size / 2; s > 0; s /= 2) {

for (inti = 0; i<size; i++) {

for (int j = i + s; j <size; j += s) {

if (array[i] >array[j]) {

int temp = array[j];

array[j] = array[i];

array[i] = temp;

}

}

}

}

}

voidDecreasingSort(intn, intmass[])

{

int new, location;

for (inti = 1; i<n; i++)

{

new = mass[i];

location = i - 1;

while (location >= 0 &&mass[location] < new)

{

mass[location + 1] = mass[location];

location = location - 1;

}

mass[location + 1] = new;

}

}

voidAscendingSort(intn, intmass[])

{

int new, location;

for (inti = 1; i<n; i++)

{

new = mass[i];

location = i - 1;

while (location >= 0 &&mass[location] > new)

{

mass[location + 1] = mass[location];

location = location - 1;

}

mass[location + 1] = new;

}

}

intcomp(constvoid\* x1, constvoid\* x2) // функциясравненияэлементовмассива

{

return (\*(int\*)x1 - \*(int\*)x2); // если результат вычитания равен 0, то числа равны, < 0: x1 <x2; > 0: x1 >x2

}

main() {

int\* a;

intN,k;

printf("Введите размер массива: ");

scanf("%d", &N);

a = (int\*)malloc(N \* sizeof(int));

srand(time(NULL));

for (inti = 0; i< N; i++)

{

a[i] = rand() % 10000;

}

DecreasingSort(N, a);

printf("\n");

clock\_t start = clock();

qsort(a, N, sizeof(int), (int(\*) (constvoid\*, constvoid\*)) comp);

clock\_t end = clock();

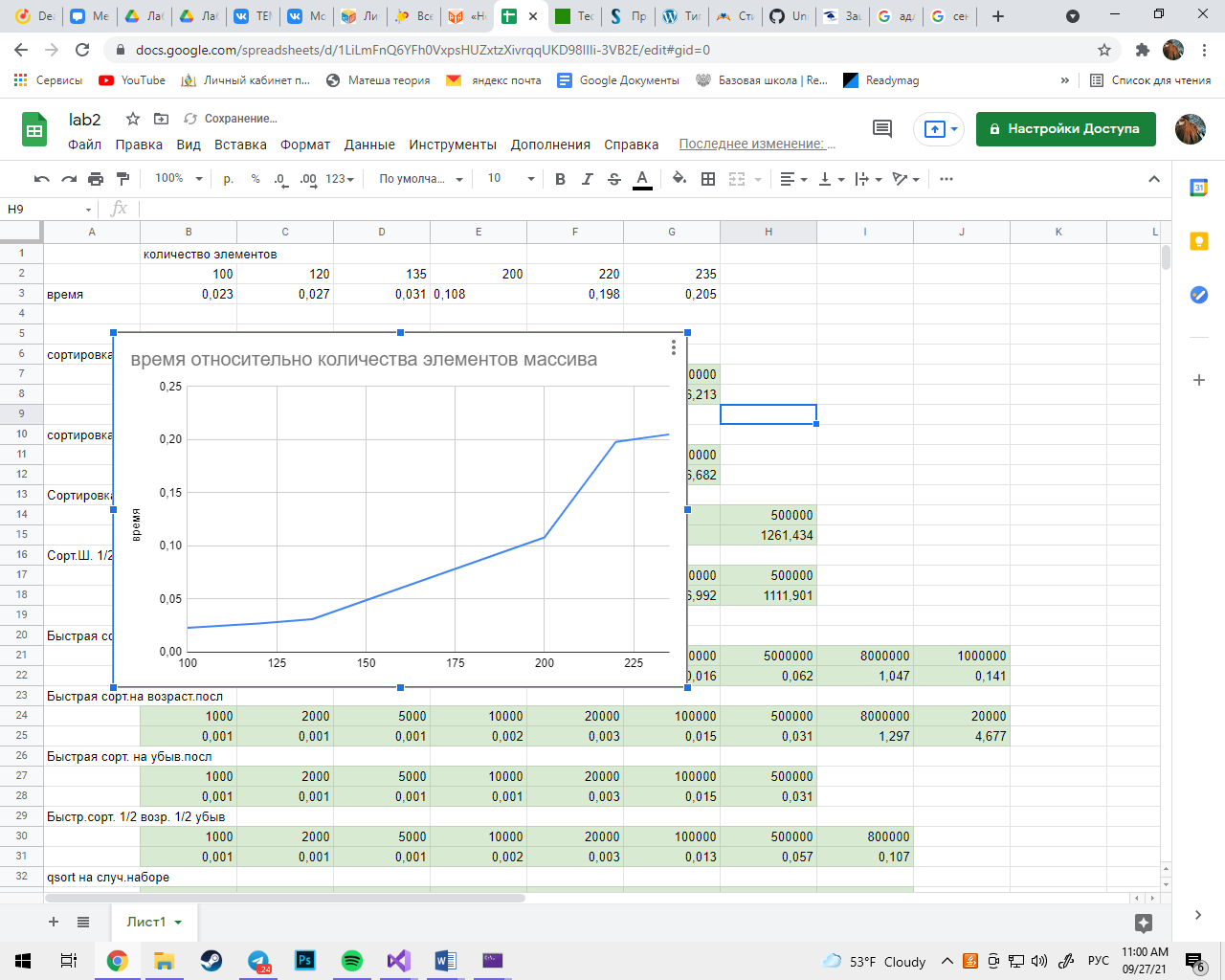
doubletime\_spent = (double)(end - start) / CLOCKS\_PER\_SEC;

printf("Сортировка случайного набора %f", time\_spent);

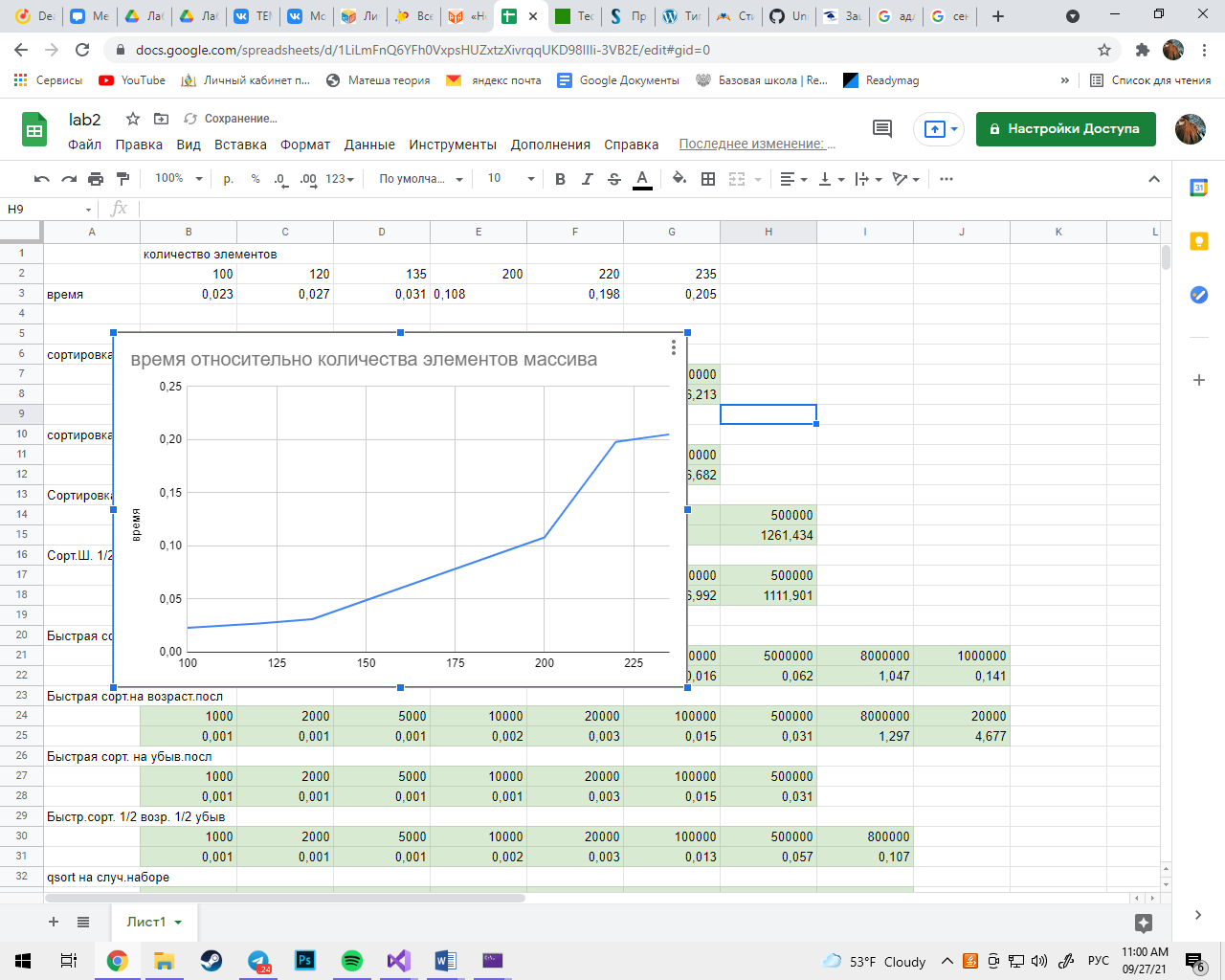
}

Задание 1:

1. O(n3)

2. 

3.

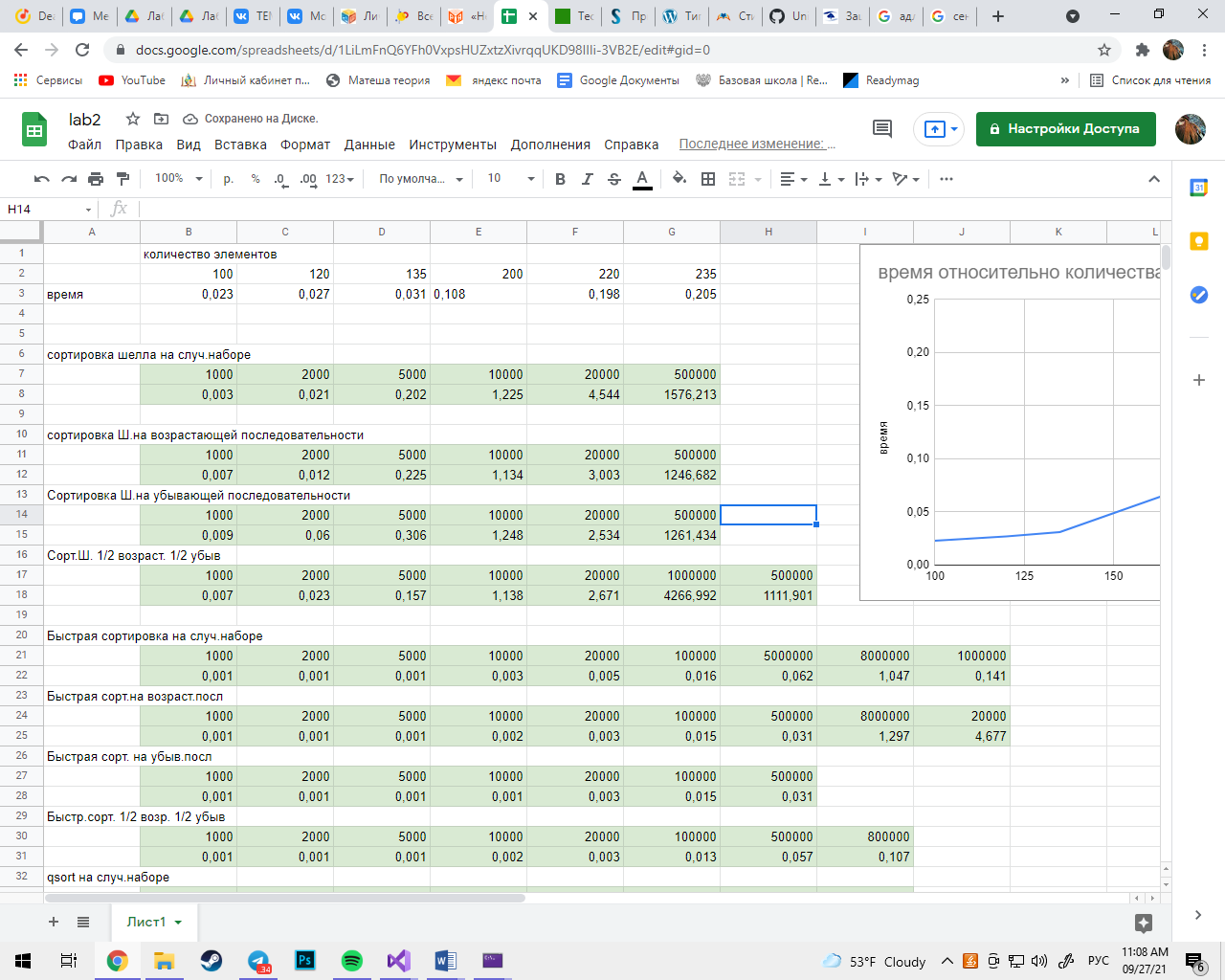


Таким образом, график подтвердил О-символику.

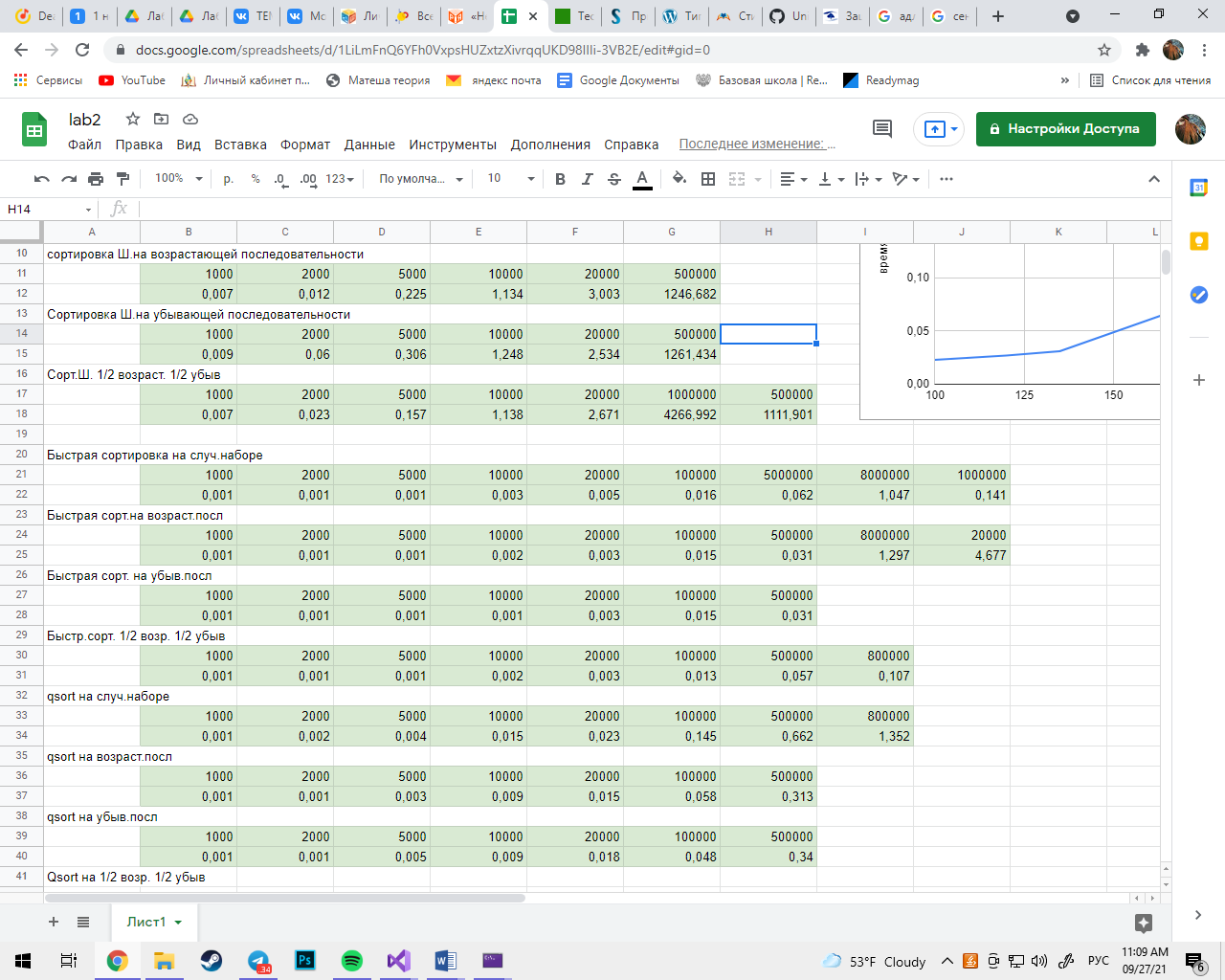
При увеличении количества элементов, время сортировки также увеличивается

Задание 2:

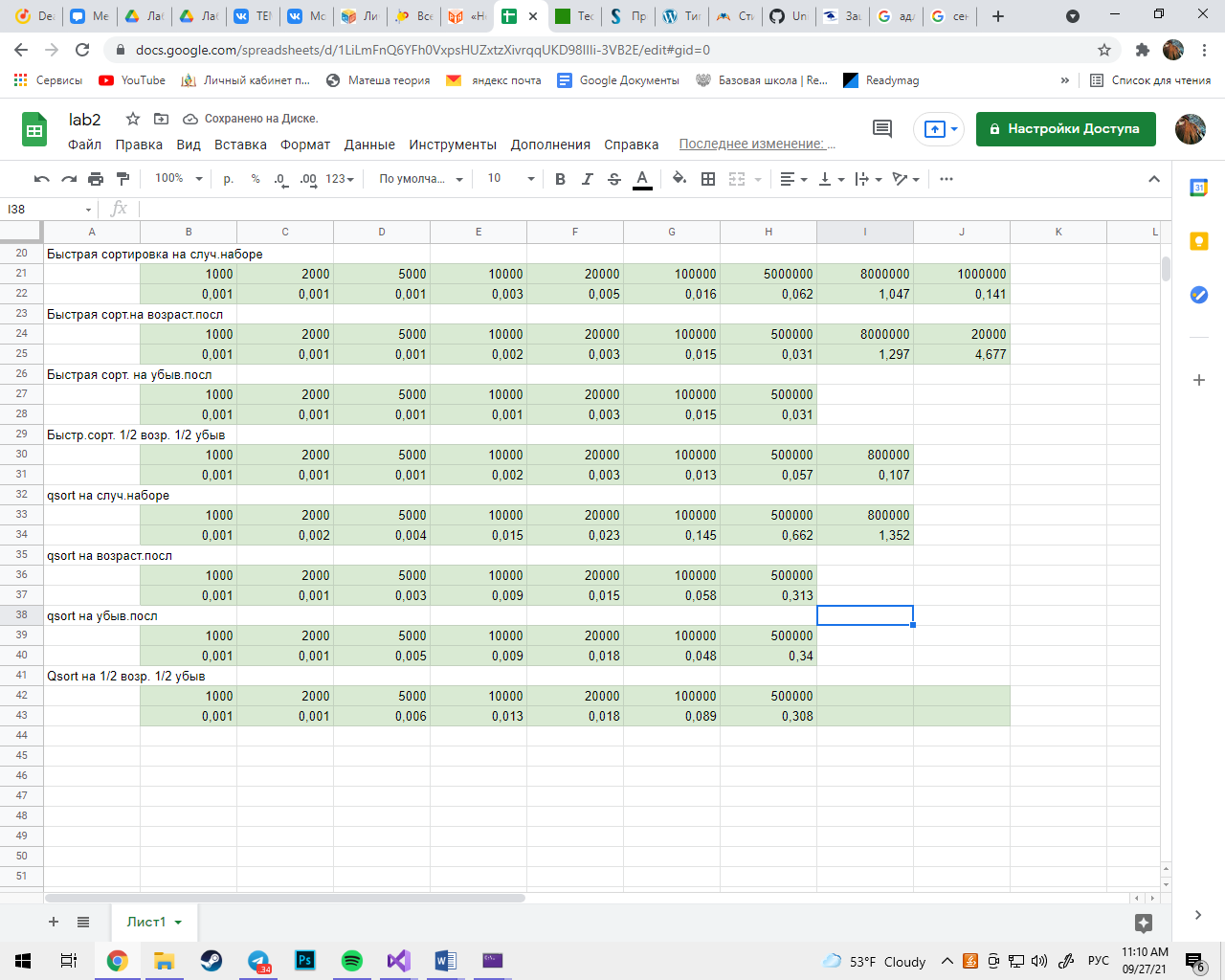
**Сортировка Шелла:**



**Быстрая сортировка:**



**qsort:**



**Вывод:**

Среди сортировок: Шелла, быстрой и qsort , самой долгой оказалась

сортировка Шелла. Быстрая сортировка показала наилучший результат по времени. Во всех случаях сортировки быстрее справлялись на уже упорядоченном по возрастанию или убыванию массиве.