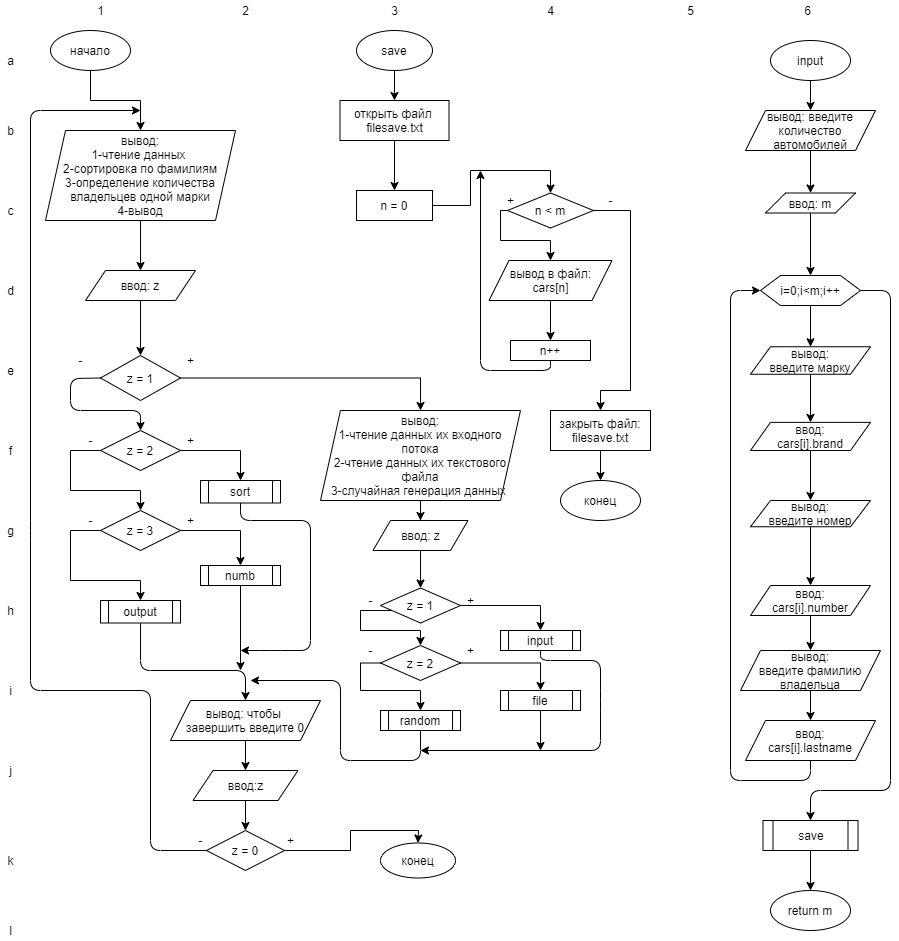
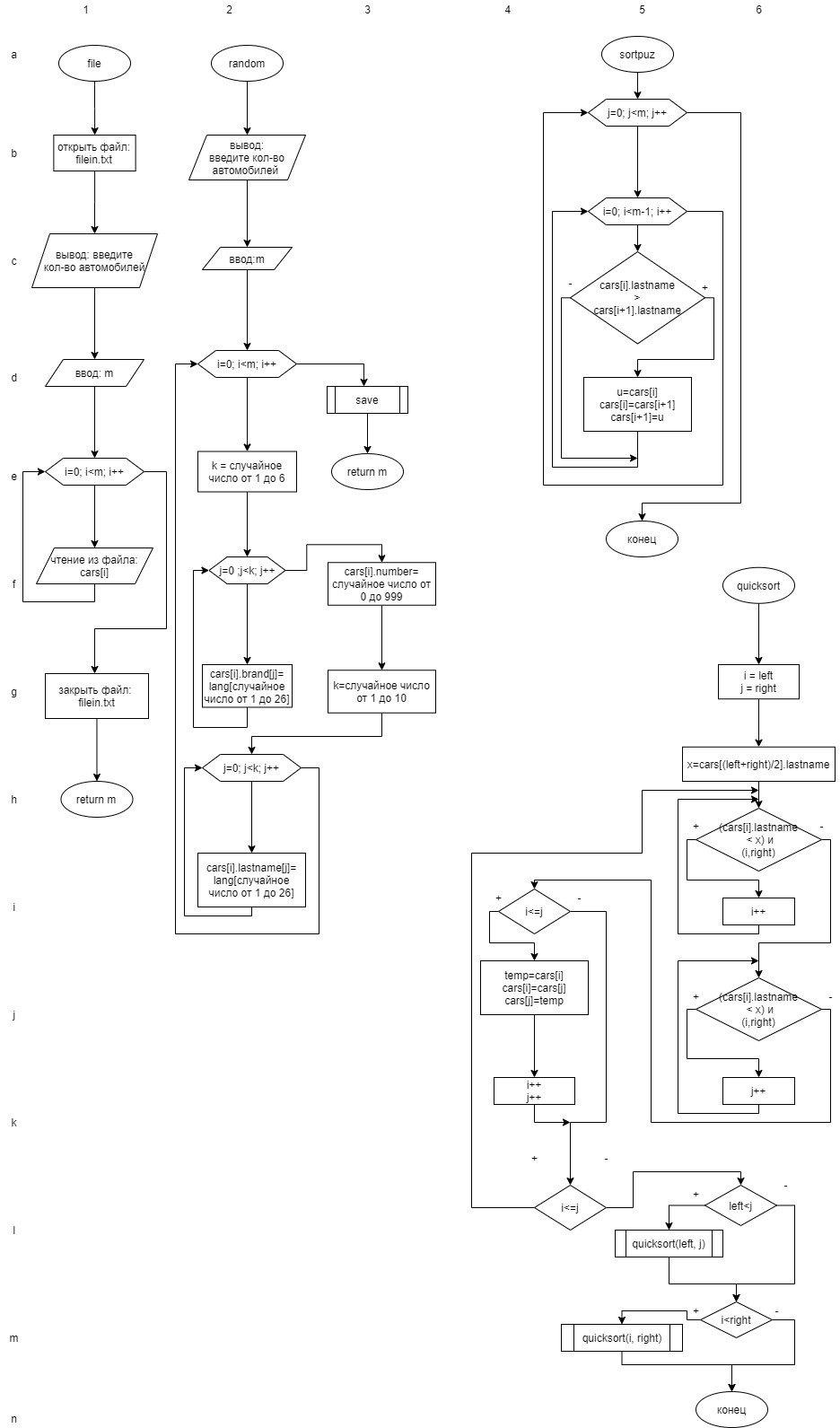
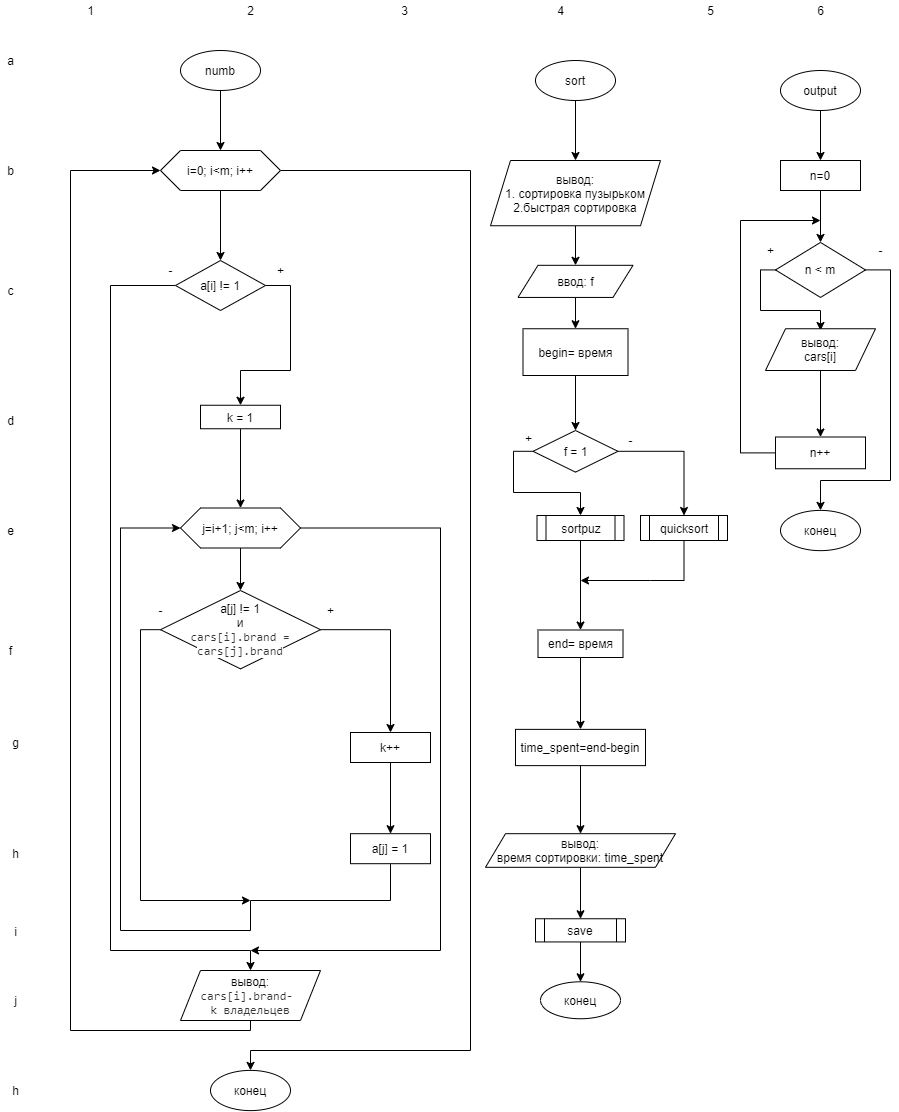
Самородов Роман Б20-513

18 Структура данных: марка автомобиля, номер автомобиля, фамилия владельца Быстрая Сортировать сведения по фамилиям владельцев. Определить количество владельцев одной марки автомобиля

**1. Блок-схема алгоритма работы основной программы и функций**

****

****

****

**2. Исходные коды всех программ.**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

struct car

{

char brand[15];

int number;

char lastname[15];

};

int save(struct car\* cars, int m)// Запись текущего состояния в файл

{

FILE \*in;

in = fopen("filesave.txt", "w");

if(in == NULL)

{

printf("Файл для сохранения текущего состояния не найден");

return 0;

}

int n = 0;

fprintf(in, "| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |\n---------------------------------------------------------------\n");

while (n < m)

{

fprintf(in, "|%3d|%18s|%18d|%19s|\n", n + 1, cars[n].brand, cars[n].number, cars[n].lastname);

n++;

}

fprintf(in, "---------------------------------------------------------------\n");

fclose(in);

return(1);

}

int input(struct car\* cars) // Чтение данных из входного потока

{

int m, i;

printf("\nВведите количество автомобилей: ");

scanf("%d",&m);

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

{

printf("Введите марку автомобиля: ");

scanf("%s",&cars[i].brand);

printf("Введите номер: ");

scanf("%d",&cars[i].number);

printf("Введите фамилию владельца: ");

scanf("%s",&cars[i].lastname);

}

save(cars, m);

return(m);

}

int file(struct car\* cars) // Чтение файлов из текстового файла

{

int m = 0, i;

FILE \*in;

in = fopen("filein.txt", "r");

if(in == NULL)

{

printf("\nФайл не найден");

return 0;

}

printf("\nВведите количество автомобилей: ");

scanf("%d",&m);

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

{

fscanf(in, "%s",&cars[i].brand);

fscanf(in, "%d",&cars[i].number);

fscanf(in, "%s",&cars[i].lastname);

}

fclose(in);

save(cars, m);

return(m);

}

int random(struct car\* cars) // Случайная генерация данных

{

int m, k;

char lang[] = "abcdefghijklmnopkrstuvwxyz";

printf("\nВведите количество автомобилей: ");

scanf("%d",&m);

srand(time(NULL));

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

{

k = rand()%6 + 1;

for (int j = 0; j < k; j++)

{

cars[i].brand[j] = lang[rand()%26];

}

cars[i].number = rand()%900 + 100;

k = rand()%10 + 1;

for (int j = 0; j < k; j++)

{

cars[i].lastname[j] = lang[rand()%26];

}

}

save(cars, m);

return(m);

}

void sortpuz(struct car\* cars, int m)// Сортировка пузырьком

{

struct car u;

for(int j = 0; j < m; j++)

{

for(int i = 0; i < m-1; i++)

{

if(strcmp(cars[i].lastname, cars[i+1].lastname)>0)

{

u = cars[i];

cars[i] = cars[i+1];

cars[i + 1] = u;

}

}

}

}

void quicksort(struct car\* cars, int left, int right)// Быстрая сортировка

{

int i = left, j = right;

char \*x;

struct car temp;

x = cars[(left+right)/2].lastname;

do

{

while((strcmp(cars[i].lastname, x) < 0) && (i < right))

i++;

while((strcmp(cars[j].lastname, x) > 0) && (j > left))

j--;

if(i <= j)

{

temp = cars[i];

cars[i] = cars[j];

cars[j] = temp;

i++;

j--;

}

} while(i <= j);

if(left < j)

quicksort(cars, left, j);

if(i < right)

quicksort(cars, i, right);

}

void sort(struct car\* cars, int m) // Сортировка

{

int f;

printf("\n1. Сортировка пузырьком\n2. Быстрая сортировка\n");

scanf("%d", &f);

clock\_t begin = clock();

if (f == 1)

sortpuz(cars, m);

else

quicksort(cars, 0 , m-1);

clock\_t end = clock();

double time\_spent = (double)(end - begin);

printf("\nВремя сортировки: %f мс\n", time\_spent);

save(cars, m);

}

void numb(struct car\* cars, int m) // Определить количество владельцев одной марки

{

int k;

int a[m];

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

{

if(a[i] != 1)

{

k = 1;

for(int j=i+1; j < m; j++)

{

if ((a[j] != 1) && (strcmp (cars[i].brand, cars[j].brand) == 0))

{

k++;

a[j] = 1;

}

}

printf("%s - %d владельцев\n", cars[i].brand, k);

}

}

}

void output(struct car\* cars, int m) // Вывод

{

int n = 0;

printf("\n| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |\n---------------------------------------------------------------\n");

while (n < m)

{

printf("|%3d|%18s|%18d|%19s|\n", n + 1, cars[n].brand, cars[n].number, cars[n].lastname);

n++;

}

printf("---------------------------------------------------------------\n");

}

void main()

{

int z, y, m, MAX = 100;

struct car\* cars;

cars = (struct car \*)malloc(sizeof(struct car) \* MAX);

do {

printf("\n1. Ввод данных\n2. Сортировка по фамилиям\n3. Определить количество владельцев одной марки\n4. Вывод\n");

scanf("%d", &z);

switch(z)

{

case 1:

printf("\n1. Чтение данных из входного потока\n2. Чтение файлов из текстового файла\n3. Случайная генерация данных\n");

scanf("%d", &y);

switch(y)

{

case 1:

m = input(cars);

break;

case 2:

m = file(cars);

break;

case 3:

m = random(cars);

}

break;

case 2:

sort(cars, m);

break;

case 3:

numb(cars, m);

break;

case 4:

output(cars, m);

}

printf("\nЧтобы завершить работу программы введите 0, иначе введите 1\n");

scanf("%d", &z);

}while (z != 0);

free(cars);

cars = NULL;

}

**3. Тестовые наборы для программы и результаты таймирования разных видов сортировки**

**Тест 1**

**Входные данные:**

tesla

456

bobrovskaya

lamborghini

365

samorodov

tesla

834

kuchebo

bmw

763

glushko

tesla

777

hramov

rover

645

apalkova

ferrari

834

hanbekova

ferrari

811

radostev

bmw

345

nazarko

bmw

363

moskovchuk

**Сортировка пузырьком:**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| rover| 645| apalkova|

| 2| tesla| 456| bobrovskaya|

| 3| bmw| 763| glushko|

| 4| ferrari| 834| hanbekova|

| 5| tesla| 777| hramov|

| 6| tesla| 834| kuchebo|

| 7| bmw| 363| moskovchuk|

| 8| bmw| 345| nazarko|

| 9| ferrari| 811| radostev|

| 10| lamborghini| 365| samorodov|

---------------------------------------------------------------

**Время сортировки 7 мс**

**Быстрая сортировка:**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| rover| 645| apalkova|

| 2| tesla| 456| bobrovskaya|

| 3| bmw| 763| glushko|

| 4| ferrari| 834| hanbekova|

| 5| tesla| 777| hramov|

| 6| tesla| 834| kuchebo|

| 7| bmw| 363| moskovchuk|

| 8| bmw| 345| nazarko|

| 9| ferrari| 811| radostev|

| 10| lamborghini| 365| samorodov|

---------------------------------------------------------------

**Время сортировки 6 мс**

**Тест 2**

**Случайная генерация данных:**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| atedki| 743| dortgova|

| 2| kesla| 187| zbjbnvekaya|

| 3| ogfpari| 296| klushko|

| 4| ztahsoi| 925| fgpbekova|

| 5| wysla| 865| btaoog|

| 6| fmsla| 109| atkhebo|

| 7| smw| 902| caanlfthuk|

| 8| imw| 744| krclsko|

| 9| zerrari| 962| elpznklzf|

| 10| jhujorghini| 306| kphzjhrblca|

| 11| edts| 474| hlurytw|

| 12| ffo| 848| vhrciclb|

| 13| xf| 192| isvsycds|

| 14| couw| 729| kjbf|

| 15| sja| 280| daviayrut|

| 16| tx| 917| vtbpru|

| 17| b| 138| iylrbine|

| 18| jnk| 941| fdracinxd|

| 19| mu| 215| xr|

| 20| fsiyt| 624| xaz|

| 21| m| 832| tnmtkwj|

| 22| aa| 844| njw|

| 23| fbznas| 515| r|

| 24| neuxyn| 641| hdjss|

| 25| sw| 661| fdol|

| 26| na| 769| gzzlod|

| 27| ldvy| 914| bbw|

| 28| mosvv| 204| kn|

| 29| znt| 319| sghg|

| 30| otolsg| 639| hl|

| 31| tbkpzk| 804| zakmtr|

| 32| n| 353| vihoztgf|

| 33| cnwtg| 885| xy|

| 34| pjkr| 872| mpnzpgwx|

| 35| ly| 920| erxtprbo|

| 36| znk| 440| giulw|

| 37| ayb| 776| a|

| 38| f| 405| bdxywor|

| 39| dk| 292| grk|

| 40| znc| 463| cvalka|

---------------------------------------------------------------

**Сортировка пузырьком:**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| ayb| 776| a|

| 2| fmsla| 109| atkhebo|

| 3| ldvy| 914| bbw|

| 4| f| 405| bdxywor|

| 5| wysla| 865| btaoog|

| 6| smw| 902| caanlfthuk|

| 7| znc| 463| cvalka|

| 8| sja| 280| daviayrut|

| 9| atedki| 743| dortgova|

| 10| zerrari| 962| elpznklzf|

| 11| ly| 920| erxtprbo|

| 12| sw| 661| fdol|

| 13| jnk| 941| fdracinxd|

| 14| ztahsoi| 925| fgpbekova|

| 15| znk| 440| giulw|

| 16| dk| 292| grk|

| 17| na| 769| gzzlod|

| 18| neuxyn| 641| hdjss|

| 19| otolsg| 639| hl|

| 20| edts| 474| hlurytw|

| 21| xf| 192| isvsycds|

| 22| b| 138| iylrbine|

| 23| couw| 729| kjbf|

| 24| ogfpari| 296| klushko|

| 25| mosvv| 204| kn|

| 26| jhujorghini| 306| kphzjhrblca|

| 27| imw| 744| krclsko|

| 28| pjkr| 872| mpnzpgwx|

| 29| aa| 844| njw|

| 30| fbznas| 515| r|

| 31| znt| 319| sghg|

| 32| m| 832| tnmtkwj|

| 33| ffo| 848| vhrciclb|

| 34| n| 353| vihoztgf|

| 35| tx| 917| vtbpru|

| 36| fsiyt| 624| xaz|

| 37| mu| 215| xr|

| 38| cnwtg| 885| xy|

| 39| tbkpzk| 804| zakmtr|

| 40| kesla| 187| zbjbnvekaya|

---------------------------------------------------------------

**Время сортировки: 48 мс**

**Тест 3**

**Случайная генерация данных:**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| osasa| 523| rmcuabo|

| 2| hnklsg| 568| ccoprz|

| 3| sfcusx| 799| uatktkehmy|

| 4| xfuh| 931| ldsav|

| 5| dpyfjti| 921| pxszredyk|

| 6| ipgcmr| 408| ihlckixp|

| 7| yw| 682| rgipyjie|

| 8| ivkkkz| 681| ifvgpjgkg|

| 9| ttanas| 230| dennxxblv|

| 10| tcfkari| 116| jriaruikk|

| 11| httla| 699| fygonvekaya|

| 12| mj| 896| nifjklbvl|

| 13| owlczk| 649| tdzowaacb|

| 14| sku| 877| srwyaaxit|

| 15| lbk| 148| nuujhavhck|

| 16| bmugv| 151| ibokdptw|

| 17| nzsvorghini| 847| kmgsvrvbrk|

| 18| vmhwad| 587| ofcykova|

| 19| risha| 328| patxtokoph|

| 20| mvgnbh| 719| wmok|

| 21| vozghe| 760| bxyabine|

| 22| oodpari| 659| lskshko|

| 23| igjiueghini| 961| abumcprblca|

| 24| dxlcari| 973| mtmscko|

| 25| jnafhw| 547| hnwkvcds|

| 26| hykmv| 509| osrdwjfz|

| 27| rvwr| 429| laznptfxsi|

| 28| zxas| 274| ptykaycon|

| 29| dhtie| 176| fngalmlb|

| 30| ydc| 394| xmikbtss|

| 31| ohdyt| 209| pkfuau|

| 32| dkpbyn| 693| nhwbnljp|

| 33| xk| 512| mcwc|

| 34| dhakw| 620| smjosm|

| 35| fugij| 471| yrpgodi|

| 36| fbpob| 901| pkfbkj|

| 37| wvoe| 419| kltemycnkp|

| 38| drwv| 714| ykzglpuo|

| 39| hfj| 721| kleyrf|

| 40| brdgw| 647| rzpsjraxi|

---------------------------------------------------------------

**Быстрая сортировка**

| № | Марка автомобиля | Номер автомобиля | Фамилия владельца |

---------------------------------------------------------------

| 1| igjiueghini| 961| abumcprblca|

| 2| vozghe| 760| bxyabine|

| 3| hnklsg| 568| ccoprz|

| 4| ttanas| 230| dennxxblv|

| 5| dhtie| 176| fngalmlb|

| 6| httla| 699| fygonvekaya|

| 7| jnafhw| 547| hnwkvcds|

| 8| bmugv| 151| ibokdptw|

| 9| ivkkkz| 681| ifvgpjgkg|

| 10| ipgcmr| 408| ihlckixp|

| 11| tcfkari| 116| jriaruikk|

| 12| hfj| 721| kleyrf|

| 13| wvoe| 419| kltemycnkp|

| 14| rvwr| 429| laznptfxsi|

| 15| mj| 896| nifjklbvl|

| 16| nzsvorghini| 847| kmgsvrvbrk|

| 17| xfuh| 931| ldsav|

| 18| xk| 512| mcwc|

| 19| dkpbyn| 693| nhwbnljp|

| 20| lbk| 148| nuujhavhck|

| 21| oodpari| 659| lskshko|

| 22| dxlcari| 973| mtmscko|

| 23| risha| 328| patxtokoph|

| 24| vmhwad| 587| ofcykova|

| 25| hykmv| 509| osrdwjfz|

| 26| fbpob| 901| pkfbkj|

| 27| ohdyt| 209| pkfuau|

| 28| zxas| 274| ptykaycon|

| 29| dpyfjti| 921| pxszredyk|

| 30| yw| 682| rgipyjie|

| 31| sku| 877| srwyaaxit|

| 32| osasa| 523| rmcuabo|

| 33| brdgw| 647| rzpsjraxi|

| 34| owlczk| 649| tdzowaacb|

| 35| sfcusx| 799| uatktkehmy|

| 36| dhakw| 620| smjosm|

| 37| mvgnbh| 719| wmok|

| 38| ydc| 394| xmikbtss|

| 39| drwv| 714| ykzglpuo|

| 40| fugij| 471| yrpgodi|

---------------------------------------------------------------

**Время сортировки: 16 мс**

**Тест 4**

**Определить количество владельцев одной марки автомобиля**

**Входные данные**:

tesla

456

bobrovskaya

lamborghini

365

samorodov

tesla

834

kuchebo

bmw

763

glushko

tesla

777

hramov

rover

645

apalkova

ferrari

834

hanbekova

ferrari

811

radostev

bmw

345

nazarko

bmw

363

moskovchuk

**Результат:**

tesla - 3 владельцев

lamborghini - 1 владельцев

bmw - 3 владельцев

rover - 1 владельцев

ferrari - 2 владельцев

|  |  |  |
| --- | --- | --- |
| **Количество автомобилей** | **Сортировка пузырьком (мс)** | **Быстрая сортировка**  **(мс)** |
| 10 | 9 | 6 |
| 20 | 16 | 9 |
| 30 | 29 | 11 |
| 40 | 46 | 15 |
| 50 | 72 | 17 |
| 60 | 100 | 21 |
| 70 | 141 | 25 |
| 80 | 172 | 26 |

**Вывод:**

Отличие во времени быстрой сортировки и сортировки пузырьком становится заметно при большом количестве данных, что подтверждает гипотезу.

**4. Valgrind**

