**Lab Description:**

\*In README file as well as the answers to the questions on lab directions

\*Not on Write up

**Question C. Graph:**

**Main:**

\*My sort.cpp is my file that has all my functions and main

**Sort.cpp:**

#include <iostream> // cout

#include <stdio.h> // srand, rand()

#include <time.h> // time functions

#include <chrono> // using crhono and time clocking algorithms

int count = 0;

void bubbleSort(int\*, int, bool); // sort the funtion using bubble sort algorithm

void selectionSort(int\*, int, bool); // sort the function using selection sort algorithm

void insertionSort(int\*, int, bool); // sort the function using insertion sort algorithm

int\* makeArray(int); // creates a dynamic array of integers

void printArray(int\*, int); // prints the array to the screen

void swap(int&, int&); // swaps two integers that are passed by reference

bool isSorted(int\*, int, bool); // tests to see if the array is sorted in the correct order

int main() {

bool ascending = true; // boolean to see which way the array is to be sorted

srand(time(NULL)); // seed the random number generator

int sizesLen = 13;

int sizes[] = {10, 20, 50, 500, 700, 1000, 5000, 7500, 12000, 25000, 50000, 60000, 100000}; // Amount of integers test cases

for (int i = 0; i < sizesLen; i++) { // Bubble sort ascending tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting bubble sort in ascending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

bubbleSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in ascending order!" << std::endl;

}

count = 0;

delete [] arr;

}

std::cout << "===============================================" << std::endl;

for (int i = 0; i < sizesLen; i++) { // Selection sort ascending tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting selection sort in ascending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

selectionSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in ascending order!" << std::endl;

}

count = 0;

delete [] arr;

}

std::cout << "===============================================" << std::endl;

for (int i = 0; i < sizesLen; i++) { // Insertion sort ascending tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting insertion sort in ascending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

insertionSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in ascending order!" << std::endl;

}

count = 0;

delete [] arr;

}

std::cout << "===============================================" << std::endl;

ascending = false; // change to descending order tests

for (int i = 0; i < sizesLen; i++) { // Bubble sort descending tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting bubble sort in descending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

bubbleSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in descending order!" << std::endl;

}

count = 0;

delete [] arr;

}

std::cout << "===============================================" << std::endl;

for (int i = 0; i < sizesLen; i++) { // Selection sort descendng tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting selection sort in descending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

selectionSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in descending order!" << std::endl;

}

count = 0;

delete [] arr;

}

std::cout << "===============================================" << std::endl;

for (int i = 0; i < sizesLen; i++) { // Insertion sort descending tests

int \*arr = makeArray(sizes[i]);

int len = sizes[i];

std::cout << "Starting insertion sort in descending order of array of " << len << " integers\n";

auto start = std::chrono::system\_clock::now();

insertionSort(arr, len, ascending);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end - start;

std::time\_t end\_time = std::chrono::system\_clock::to\_time\_t(end);

std::cout << "Finished at " << std::ctime(&end\_time) << "Elapsed time: " << elapsed\_seconds.count() << "s\n";

std::cout << "Amount of swaps: " << count << std::endl;

if (isSorted(arr, len, ascending)) {

std::cout << "Sorted in descending order!" << std::endl;

}

count = 0;

delete [] arr;

}

return 0;

}

/\*

\* isSorted Function:

\* Takes a a dynamic array, a length of that array, and whether the array is supposed to be sorted

\* in ascending or descending order and then determines if the array is sorted correctly

\*/

bool isSorted(int\* arr, int length, bool ascending) {

if (ascending) {

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

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

return false;

}

}

return true;

} else {

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

if (arr[i] < arr[i + 1]) {

return false;

}

}

return true;

}

}

/\*

\* makeArray Function:

\* Takes an int that is the length of an array and then dynamically creates that array and returns

\* a pointer to that array

\*/

int\* makeArray(int len) {

int \*rtn = new int[len];

for (int i = 0; i < len; i++) {

rtn[i] = 1 + rand() % 100; // from 1 to 100

}

return rtn;

}

/\*

\* swap Function:

\* Takes two array elements by referene and swaps them

\*/

void swap(int& prev, int& next) {

int temp = prev;

prev = next;

next = temp;

count++; // remember to make it 0 after each test

}

/\*

\* bubbleSort Function:

\* Takes a dynamic array, the length of the array, and whether the array is to be sorted in ascending

\* or descending and then sorts the array

\*/

void bubbleSort(int\* arr, int length, bool ascending) {

bool didSwap = true;

while (didSwap) {

didSwap = false;

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

if (ascending) {

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

swap(arr[i], arr[i + 1]);

didSwap = true;

}

} else {

if (arr[i] < arr[i + 1]) {

swap(arr[i], arr[i + 1]);

didSwap = true;

}

}

}

}

}

/\*

\* selectionSort Function:

\* Takes a dynamic array, the length of the array, and whether the array is to be sorted in ascending

\* or descending order and then sorts the array

\*/

void selectionSort(int\* arr, int length, bool ascending) {

int min;

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

min = i;

for (int j = i + 1; j < length; j++) {

if (ascending) {

if (arr[j] < arr[min]) {

min = j;

}

} else {

if (arr[j] > arr[min]) {

min = j;

}

}

}

swap(arr[i], arr[min]);

}

}

/\*

\* insertionSort Function:

\* Takes a dynamic array, the length of the array, and whether the array is to be sorted in ascending

\* or descending order and then sorts the array

\*/

void insertionSort(int\* arr, int length, bool ascending) {

int j;

for (int i = 1; i < length; i++) {

j = i;

if (ascending) {

while ((j > 0) && (arr[j] < arr[j - 1])) {

swap(arr[j], arr[j - 1]);

j = j - 1;

}

} else {

while ((j > 0) && (arr[j] > arr[j - 1])) {

swap(arr[j], arr[j - 1]);

j = j - 1;

}

}

}

}

/\*

\* printArray Function:

\* Takes a dynamic array and the length of the array and prints all of the contents of the array

\* to the screen

\*/

void printArray(int\* arr, int length) {

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

printf("%d ", arr[i]);

}

puts("====================");

}

**Sample Output:**

Starting bubble sort in ascending order of array of 10 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 7.93e-07s

Amount of swaps: 21

Sorted in ascending order!

Starting bubble sort in ascending order of array of 20 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 2.811e-06s

Amount of swaps: 88

Sorted in ascending order!

Starting bubble sort in ascending order of array of 50 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 1.5647e-05s

Amount of swaps: 679

Sorted in ascending order!

Starting bubble sort in ascending order of array of 500 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 0.00122648s

Amount of swaps: 59801

Sorted in ascending order!

Starting bubble sort in ascending order of array of 700 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 0.00250836s

Amount of swaps: 121769

Sorted in ascending order!

Starting bubble sort in ascending order of array of 1000 integers

Finished at Wed Oct 9 11:08:50 2019

Elapsed time: 0.0085739s

Amount of swaps: 248832

Sorted in ascending order!

Starting bubble sort in ascending order of array of 5000 integers

Finished at Wed Oct 9 11:08:51 2019

Elapsed time: 0.158688s

Amount of swaps: 6212462

Sorted in ascending order!

Starting bubble sort in ascending order of array of 7500 integers

Finished at Wed Oct 9 11:08:51 2019

Elapsed time: 0.321245s

Amount of swaps: 13867834

Sorted in ascending order!

Starting bubble sort in ascending order of array of 12000 integers

Finished at Wed Oct 9 11:08:52 2019

Elapsed time: 0.764296s

Amount of swaps: 35635023

Sorted in ascending order!

Starting bubble sort in ascending order of array of 25000 integers

Finished at Wed Oct 9 11:08:55 2019

Elapsed time: 3.20349s

Amount of swaps: 155861860

Sorted in ascending order!

Starting bubble sort in ascending order of array of 50000 integers

Finished at Wed Oct 9 11:09:08 2019

Elapsed time: 12.6572s

Amount of swaps: 619341268

Sorted in ascending order!

Starting bubble sort in ascending order of array of 60000 integers

Finished at Wed Oct 9 11:09:26 2019

Elapsed time: 18.3723s

Amount of swaps: 890269911

Sorted in ascending order!

Starting bubble sort in ascending order of array of 100000 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 50.5987s

Amount of swaps: -1816001663

Sorted in ascending order!

===============================================

Starting selection sort in ascending order of array of 10 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 5.46e-07s

Amount of swaps: 9

Sorted in ascending order!

Starting selection sort in ascending order of array of 20 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 1.055e-06s

Amount of swaps: 19

Sorted in ascending order!

Starting selection sort in ascending order of array of 50 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 4.486e-06s

Amount of swaps: 49

Sorted in ascending order!

Starting selection sort in ascending order of array of 500 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.000279215s

Amount of swaps: 499

Sorted in ascending order!

Starting selection sort in ascending order of array of 700 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.000531035s

Amount of swaps: 699

Sorted in ascending order!

Starting selection sort in ascending order of array of 1000 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.0010807s

Amount of swaps: 999

Sorted in ascending order!

Starting selection sort in ascending order of array of 5000 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.0535509s

Amount of swaps: 4999

Sorted in ascending order!

Starting selection sort in ascending order of array of 7500 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.126358s

Amount of swaps: 7499

Sorted in ascending order!

Starting selection sort in ascending order of array of 12000 integers

Finished at Wed Oct 9 11:10:17 2019

Elapsed time: 0.275966s

Amount of swaps: 11999

Sorted in ascending order!

Starting selection sort in ascending order of array of 25000 integers

Finished at Wed Oct 9 11:10:18 2019

Elapsed time: 0.806314s

Amount of swaps: 24999

Sorted in ascending order!

Starting selection sort in ascending order of array of 50000 integers

Finished at Wed Oct 9 11:10:21 2019

Elapsed time: 2.83686s

Amount of swaps: 49999

Sorted in ascending order!

Starting selection sort in ascending order of array of 60000 integers

Finished at Wed Oct 9 11:10:25 2019

Elapsed time: 4.69175s

Amount of swaps: 59999

Sorted in ascending order!

Starting selection sort in ascending order of array of 100000 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 11.5116s

Amount of swaps: 99999

Sorted in ascending order!

===============================================

Starting insertion sort in ascending order of array of 10 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 3.7e-07s

Amount of swaps: 12

Sorted in ascending order!

Starting insertion sort in ascending order of array of 20 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 9.24e-07s

Amount of swaps: 88

Sorted in ascending order!

Starting insertion sort in ascending order of array of 50 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 3.906e-06s

Amount of swaps: 502

Sorted in ascending order!

Starting insertion sort in ascending order of array of 500 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 0.000635248s

Amount of swaps: 60546

Sorted in ascending order!

Starting insertion sort in ascending order of array of 700 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 0.000848772s

Amount of swaps: 110835

Sorted in ascending order!

Starting insertion sort in ascending order of array of 1000 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 0.00174017s

Amount of swaps: 241681

Sorted in ascending order!

Starting insertion sort in ascending order of array of 5000 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 0.0789309s

Amount of swaps: 6114412

Sorted in ascending order!

Starting insertion sort in ascending order of array of 7500 integers

Finished at Wed Oct 9 11:10:37 2019

Elapsed time: 0.202202s

Amount of swaps: 13946497

Sorted in ascending order!

Starting insertion sort in ascending order of array of 12000 integers

Finished at Wed Oct 9 11:10:38 2019

Elapsed time: 0.39265s

Amount of swaps: 35706232

Sorted in ascending order!

Starting insertion sort in ascending order of array of 25000 integers

Finished at Wed Oct 9 11:10:39 2019

Elapsed time: 1.21915s

Amount of swaps: 154615267

Sorted in ascending order!

Starting insertion sort in ascending order of array of 50000 integers

Finished at Wed Oct 9 11:10:44 2019

Elapsed time: 4.71112s

Amount of swaps: 620930234

Sorted in ascending order!

Starting insertion sort in ascending order of array of 60000 integers

Finished at Wed Oct 9 11:10:50 2019

Elapsed time: 6.7361s

Amount of swaps: 891467187

Sorted in ascending order!

Starting insertion sort in ascending order of array of 100000 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 18.5674s

Amount of swaps: -1823242680

Sorted in ascending order!

===============================================

Starting bubble sort in descending order of array of 10 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 6.22e-07s

Amount of swaps: 22

Sorted in descending order!

Starting bubble sort in descending order of array of 20 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 1.996e-06s

Amount of swaps: 117

Sorted in descending order!

Starting bubble sort in descending order of array of 50 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 1.0312e-05s

Amount of swaps: 587

Sorted in descending order!

Starting bubble sort in descending order of array of 500 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 0.00089076s

Amount of swaps: 61851

Sorted in descending order!

Starting bubble sort in descending order of array of 700 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 0.00178783s

Amount of swaps: 123677

Sorted in descending order!

Starting bubble sort in descending order of array of 1000 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 0.00356319s

Amount of swaps: 248627

Sorted in descending order!

Starting bubble sort in descending order of array of 5000 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 0.208719s

Amount of swaps: 6249804

Sorted in descending order!

Starting bubble sort in descending order of array of 7500 integers

Finished at Wed Oct 9 11:11:09 2019

Elapsed time: 0.348841s

Amount of swaps: 14026522

Sorted in descending order!

Starting bubble sort in descending order of array of 12000 integers

Finished at Wed Oct 9 11:11:10 2019

Elapsed time: 0.787016s

Amount of swaps: 35868692

Sorted in descending order!

Starting bubble sort in descending order of array of 25000 integers

Finished at Wed Oct 9 11:11:13 2019

Elapsed time: 3.09812s

Amount of swaps: 154499209

Sorted in descending order!

Starting bubble sort in descending order of array of 50000 integers

Finished at Wed Oct 9 11:11:25 2019

Elapsed time: 12.163s

Amount of swaps: 621229163

Sorted in descending order!

Starting bubble sort in descending order of array of 60000 integers

Finished at Wed Oct 9 11:11:43 2019

Elapsed time: 17.6205s

Amount of swaps: 892544774

Sorted in descending order!

Starting bubble sort in descending order of array of 100000 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 48.8125s

Amount of swaps: -1823664250

Sorted in descending order!

===============================================

Starting selection sort in descending order of array of 10 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 5.3e-07s

Amount of swaps: 9

Sorted in descending order!

Starting selection sort in descending order of array of 20 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 1.264e-06s

Amount of swaps: 19

Sorted in descending order!

Starting selection sort in descending order of array of 50 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 4.647e-06s

Amount of swaps: 49

Sorted in descending order!

Starting selection sort in descending order of array of 500 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.000270767s

Amount of swaps: 499

Sorted in descending order!

Starting selection sort in descending order of array of 700 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.000580781s

Amount of swaps: 699

Sorted in descending order!

Starting selection sort in descending order of array of 1000 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.00118945s

Amount of swaps: 999

Sorted in descending order!

Starting selection sort in descending order of array of 5000 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.0584546s

Amount of swaps: 4999

Sorted in descending order!

Starting selection sort in descending order of array of 7500 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.125762s

Amount of swaps: 7499

Sorted in descending order!

Starting selection sort in descending order of array of 12000 integers

Finished at Wed Oct 9 11:12:32 2019

Elapsed time: 0.291581s

Amount of swaps: 11999

Sorted in descending order!

Starting selection sort in descending order of array of 25000 integers

Finished at Wed Oct 9 11:12:33 2019

Elapsed time: 0.787142s

Amount of swaps: 24999

Sorted in descending order!

Starting selection sort in descending order of array of 50000 integers

Finished at Wed Oct 9 11:12:36 2019

Elapsed time: 2.85242s

Amount of swaps: 49999

Sorted in descending order!

Starting selection sort in descending order of array of 60000 integers

Finished at Wed Oct 9 11:12:40 2019

Elapsed time: 4.01456s

Amount of swaps: 59999

Sorted in descending order!

Starting selection sort in descending order of array of 100000 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 11.1373s

Amount of swaps: 99999

Sorted in descending order!

===============================================

Starting insertion sort in descending order of array of 10 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 5.38e-07s

Amount of swaps: 32

Sorted in descending order!

Starting insertion sort in descending order of array of 20 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 1.048e-06s

Amount of swaps: 110

Sorted in descending order!

Starting insertion sort in descending order of array of 50 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 4.796e-06s

Amount of swaps: 577

Sorted in descending order!

Starting insertion sort in descending order of array of 500 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 0.000405406s

Amount of swaps: 61350

Sorted in descending order!

Starting insertion sort in descending order of array of 700 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 0.000900782s

Amount of swaps: 124409

Sorted in descending order!

Starting insertion sort in descending order of array of 1000 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 0.00169918s

Amount of swaps: 246507

Sorted in descending order!

Starting insertion sort in descending order of array of 5000 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 0.0786632s

Amount of swaps: 6251047

Sorted in descending order!

Starting insertion sort in descending order of array of 7500 integers

Finished at Wed Oct 9 11:12:51 2019

Elapsed time: 0.168515s

Amount of swaps: 13872225

Sorted in descending order!

Starting insertion sort in descending order of array of 12000 integers

Finished at Wed Oct 9 11:12:52 2019

Elapsed time: 0.309111s

Amount of swaps: 35810317

Sorted in descending order!

Starting insertion sort in descending order of array of 25000 integers

Finished at Wed Oct 9 11:12:53 2019

Elapsed time: 1.21445s

Amount of swaps: 155134945

Sorted in descending order!

Starting insertion sort in descending order of array of 50000 integers

Finished at Wed Oct 9 11:12:57 2019

Elapsed time: 4.46015s

Amount of swaps: 618143971

Sorted in descending order!

Starting insertion sort in descending order of array of 60000 integers

Finished at Wed Oct 9 11:13:04 2019

Elapsed time: 6.37232s

Amount of swaps: 886193458

Sorted in descending order!

Starting insertion sort in descending order of array of 100000 integers

Finished at Wed Oct 9 11:13:22 2019

Elapsed time: 17.7423s

Amount of swaps: -1818331417

Sorted in descending order!