Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Выполнил студент группы КС-36 Кошкарев Иван Михайлович

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/KoshkarevIM\_36\_ALG/tree/main/Lab1

Приняли: Крашенинников Роман Сергеевич

Дата сдачи: 17.02

Оглавление

[Описание задачи. 2](#_Toc63548272)

[Описание метода/модели. 2](#_Toc63548273)

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

Изучить способ анализа алгоритма, связанного со временем. Рассмотреть наихудшие, лучшее и среднее время для этого алгоритма и соотнести его с известным для алгоритма показателем эффективности "О"

# Описание метода/модели.

Был выбран алгоритм сортировки выбором.

Описание алгоритма:

1) Ищем наименьший элемент неотсортированного массива

2) Меняем местами 1 элемент неотсортированного массива и найденный наименьший

3) Ищем наименьший элемент неотсортированного массива (игнорируя отсортированную часть массива)

4) К шагу 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сложность сортировки:

Худший случай O(n^2)

Лучший случай Ω(n^2)

Средний случай Θ(n^2)

Максимально число обменов (n – 1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Алгоритм пространственной сложностью О(1)

Алгоритм является **неустойчивым**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Преимущества данного метода является количество замен, которое всегда равняется N, а также использование памяти с пространственной сложностью O(1)

# Выполнение задачи.

Для реализации алгоритма сортировки я использовал язык С++ из-за его скорости работы. А также язык Python для анализа данных.

Реализация сортировки:

// Функция сортировки выбором

void selectionSort(vector<double>& arr, int& passes, int& swaps) {

int n = arr.size();

passes = 0; // проходы по массиву

swaps = 0; // количество обменов

for (int i = 0; i < n - 2; i++) {

int min = i; // Индекс минимального элемента

++passes; // Считаем проходы

// Находим минимальный элемент в оставшейся части массива

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

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

min = j;

}

}

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

++swaps; // Считаем обмены

}

}

Реализация генерации массива случайных чисел:

// Генерация массива случайных чисел

vector<double> genRandArr(int size) {

vector<double> arr(size);

// Инициализация генератора случайных чисел

mt19937 engine(time(0));

uniform\_real\_distribution<double> gen(-1.0, 1.0);

// Заполняем массив случайными числами

for (auto& el : arr) {

el = gen(engine);

}

return arr;

}

Реализация проведения серий сортировок массива величиной N и записи собранных данных в файл:

// Запуск серии тестов

void runTestSeries(int arraySize, int seriesLen, ofstream& outFile) {

vector<double> times;

vector<int> passesList;

vector<int> swapsList;

//начинаем тесты

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

//создаем массив со случайными числами

auto arr = genRandArr(arraySize);

int passes = 0, swaps = 0;

//засекаем время

auto start = chrono::high\_resolution\_clock::now();

//сортировка

selectionSort(arr, passes, swaps);

//останавливаем таймер

auto end = chrono::high\_resolution\_clock::now();

//рассчитываем продолжительность

chrono::duration<double> period = end - start;

//добавляем данные в массив

times.push\_back(period.count());

passesList.push\_back(passes);

swapsList.push\_back(swaps);

cout << "Размер м: " << arraySize << ", Проходов: " << passes << ", Обменов: " << swaps << ", Время: " << period.count() << " сек\n";

}

// Вычисляем среднее, наилучшее и наихудшее время

double avgTime = accumulate(times.begin(), times.end(), 0.0) / seriesLen;

cout << "Среднее время для сортировки массива величиной " << arraySize << ": " << avgTime << " сек\n\n";

double bestTime = \*min\_element(times.begin(), times.end());

double worstTime = \*max\_element(times.begin(), times.end());

// Вычисляем среднее количество проходов и обменов

double avgPasses = accumulate(passesList.begin(), passesList.end(), 0.0) / seriesLen;

double avgSwaps = accumulate(swapsList.begin(), swapsList.end(), 0.0) / seriesLen;

// Записываем данные в файл

outFile << arraySize << "," << avgTime << "," << bestTime << "," << worstTime << "," << avgPasses << "," << avgSwaps << "\n";

}

Реализация инициализации файла и запуска тестов массивов разной величины:

void saveDataToFile(const vector<int>& arraySizes, int seriesLen) {

ofstream outFile("sorting\_data.csv");

// Заголовок файла

outFile << "ArraySize,AvgTime,BestTime,WorstTime,AvgPasses,AvgSwaps\n";

for (int size : arraySizes) {

cout << "Сортировка массивов величиной: " << size << "\n";

runTestSeries(size, seriesLen, outFile);

}

outFile.close();

cout << "Данные сохранены в файл sorting\_data.csv\n";

}

Реализация расчета "с" и построения графиков на языке Python:

# Загрузка данных из файла

data = pd.read\_csv("sorting\_data.csv")

# Размеры массивов

array\_sizes = data["ArraySize"]

# Времена выполнения

avg\_times = data["AvgTime"]

best\_times = data["BestTime"]

worst\_times = data["WorstTime"]

# Среднее количество обменов и проходов

avg\_swaps = data["AvgSwaps"]

avg\_passes = data["AvgPasses"]

# Функция для вычисления O(N^2) с подобранной константой c

def O\_N(n, c):

return c \* n\*\*2

# Подбор константы c для O(N^2)

# Найдем такое c, чтобы график O(N^2) был выше графика наихудшего времени при N > 1000

c = max(worst\_times / (array\_sizes\*\*2))

# Совмещённый график наихудшего времени и O(N^2)

plt.figure(figsize=(10, 6))

plt.plot(array\_sizes, worst\_times, label="Наихудшее время", marker="o")

plt.plot(array\_sizes, O\_N(array\_sizes, c), label=f"O(N^2), c={c:.2e}", linestyle="--")

plt.xlabel("Размер массива")

plt.ylabel("Время (сек)")

plt.title("Совмещённый график наихудшего времени и O(N^2)")

plt.legend()

plt.grid()

plt.show()

# Совмещённый график среднего, наилучшего и наихудшего времени

plt.figure(figsize=(10, 6))

plt.plot(array\_sizes, avg\_times, label="Среднее время", marker="o")

plt.plot(array\_sizes, best\_times, label="Наилучшее время", marker="o")

plt.plot(array\_sizes, worst\_times, label="Наихудшее время", marker="o")

plt.xlabel("Размер массива")

plt.ylabel("Время (сек)")

plt.title("Совмещённый график времени выполнения")

plt.legend()

plt.grid()

plt.show()

# График среднего количества обменов

plt.figure(figsize=(10, 6))

plt.plot(array\_sizes, avg\_swaps, label="Среднее количество обменов", marker="o", color="green")

plt.xlabel("Размер массива")

plt.ylabel("Количество обменов")

plt.title("График среднего количества обменов")

plt.legend()

plt.grid()

plt.show()

# График среднего количества проходов

plt.figure(figsize=(10, 6))

plt.plot(array\_sizes, avg\_passes, label="Среднее количество проходов", marker="o", color="purple")

plt.xlabel("Размер массива")

plt.ylabel("Количество проходов")

plt.title("График среднего количества проходов")

plt.legend()

plt.grid()

plt.show()

***Результаты:***

Сортировка массивов величиной: 1000

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0027389 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0024928 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025029 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025114 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0029359 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.002498 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0033648 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0028271 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025045 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0037531 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.002495 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0027042 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025487 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025469 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0027489 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0025134 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0034602 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0026434 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.002666 сек

Размер м: 1000, Проходов: 998, Обменов: 998, Время: 0.0028395 сек

Среднее время для сортировки массива величиной 1000: 0.00276478 сек

Сортировка массивов величиной: 2000

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0105287 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0113566 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0119429 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0114247 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0126988 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0112068 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0110822 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0117973 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0107279 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0101691 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0105793 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0112775 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0107609 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0113037 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0121807 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0114295 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0117063 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0138543 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0109222 сек

Размер м: 2000, Проходов: 1998, Обменов: 1998, Время: 0.0119114 сек

Среднее время для сортировки массива величиной 2000: 0.011443 сек

Сортировка массивов величиной: 4000

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0436684 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0441202 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0429535 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0464647 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0427759 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0398094 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0433547 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0421804 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0433558 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0427821 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0482419 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0425124 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0431954 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0419964 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0465706 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.045794 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0431478 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0461816 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0423246 сек

Размер м: 4000, Проходов: 3998, Обменов: 3998, Время: 0.0412287 сек

Среднее время для сортировки массива величиной 4000: 0.0436329 сек

Сортировка массивов величиной: 8000

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.157562 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.159524 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.158743 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.164584 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.160301 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.159012 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.151505 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.154876 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.168374 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.164584 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.155038 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.164806 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.214425 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.154581 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.157899 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.161472 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.157469 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.158643 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.160841 сек

Размер м: 8000, Проходов: 7998, Обменов: 7998, Время: 0.154204 сек

Среднее время для сортировки массива величиной 8000: 0.161922 сек

Сортировка массивов величиной: 16000

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.630755 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.616127 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.656072 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.626377 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.618123 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.626825 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.625417 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.622599 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.622141 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.6192 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.61814 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.625849 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.617644 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.622437 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.619685 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.616079 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.618395 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.613305 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.628553 сек

Размер м: 16000, Проходов: 15998, Обменов: 15998, Время: 0.617433 сек

Среднее время для сортировки массива величиной 16000: 0.623058 сек

Сортировка массивов величиной: 32000

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.50767 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.50383 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.50195 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.47972 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.52677 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.47781 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.49811 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.5239 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.50417 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.49041 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.48824 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.49857 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.5593 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.4911 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.48344 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.48885 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.52265 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.50976 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.51147 сек

Размер м: 32000, Проходов: 31998, Обменов: 31998, Время: 2.52528 сек

Среднее время для сортировки массива величиной 32000: 2.50465 сек

Сортировка массивов величиной: 64000

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.7843 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.80412 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.99581 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.76769 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.77344 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.78604 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.94496 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.78681 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.75522 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.9129 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.76341 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.8352 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.81401 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.7853 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.76778 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.9305 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.75387 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.80644 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 10.0674 сек

Размер м: 64000, Проходов: 63998, Обменов: 63998, Время: 9.97253 сек

Среднее время для сортировки массива величиной 64000: 9.84038 сек

Сортировка массивов величиной: 128000

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 38.9606 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.2779 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.5841 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.2229 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.6295 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.3743 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.0253 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.0638 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.2207 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.0405 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.0851 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.1756 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 38.8477 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.01 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.0833 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.123 сек

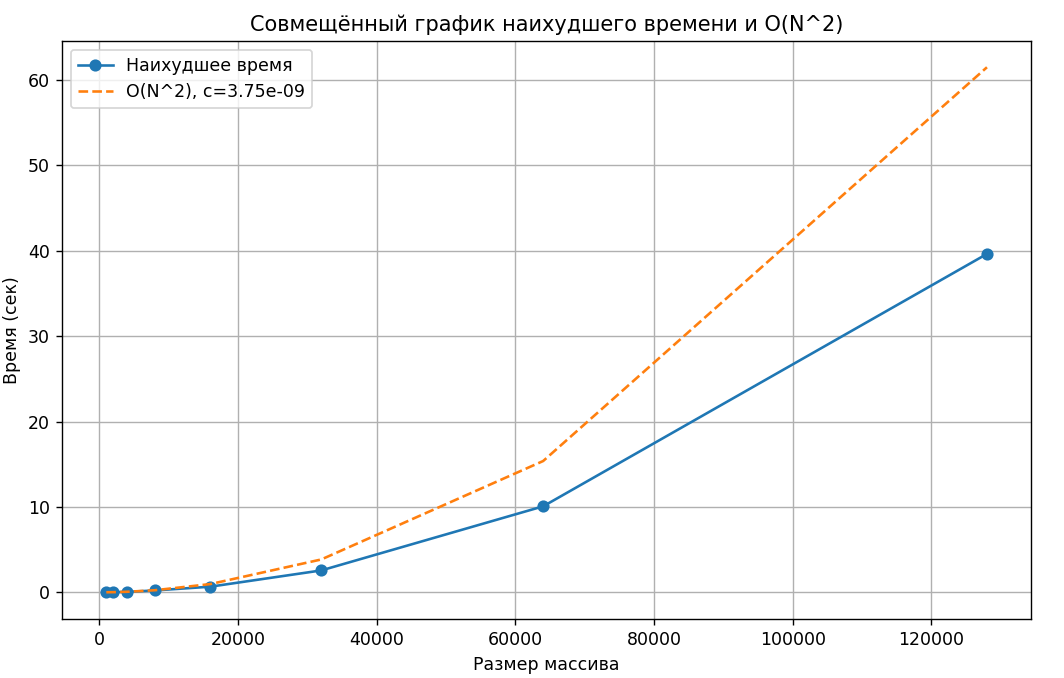
Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.1254 сек

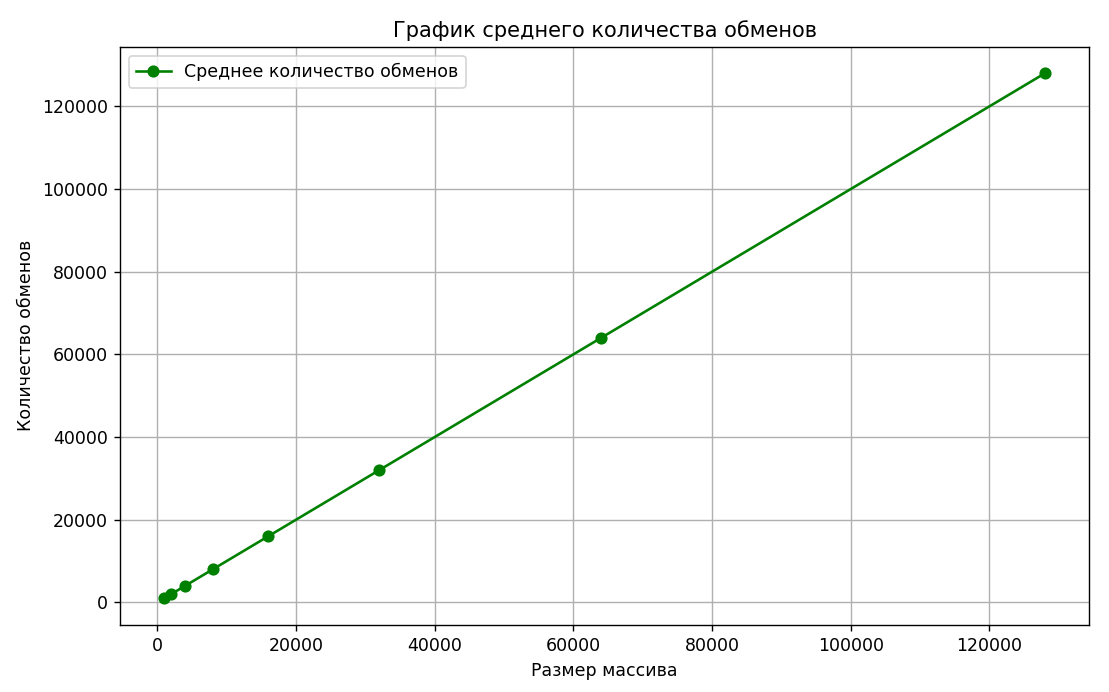
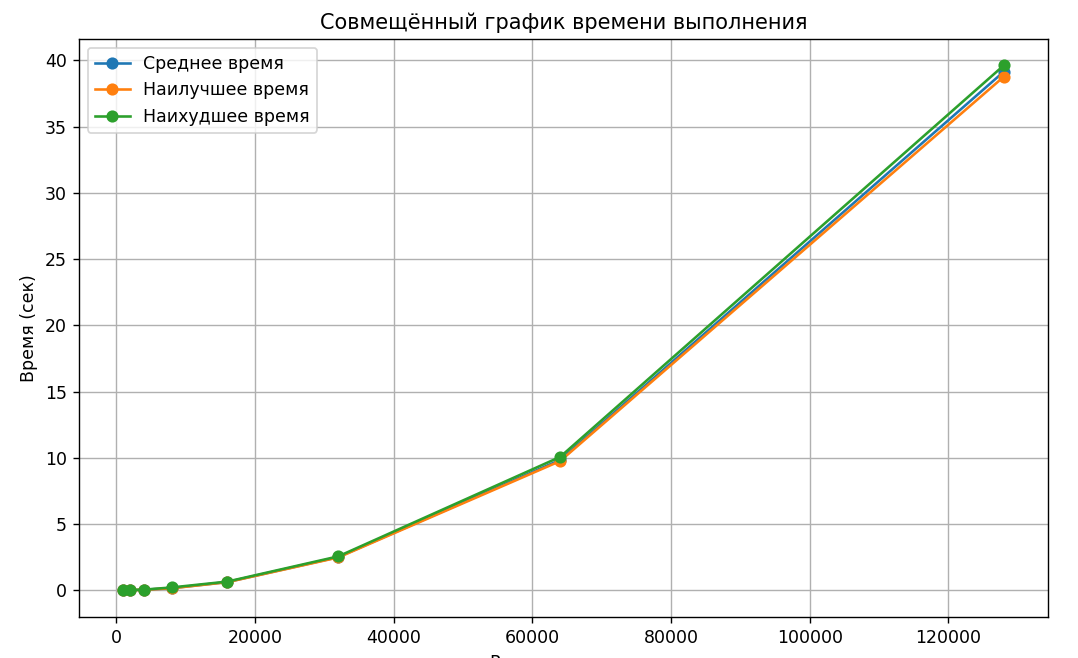
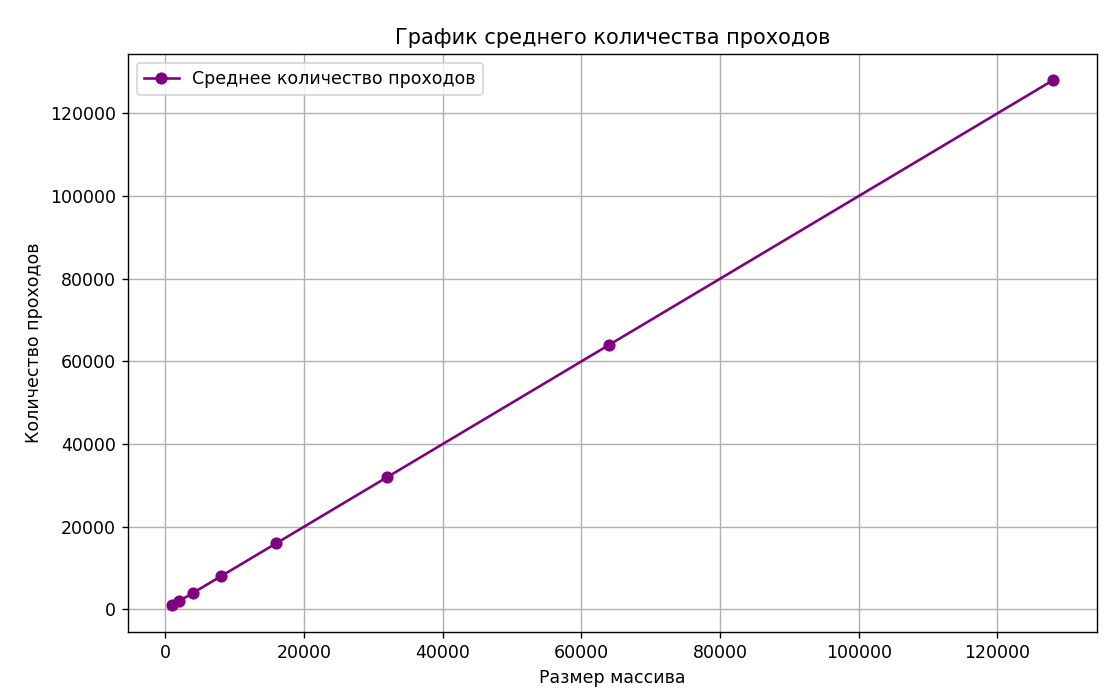
Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 39.5233 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 38.9582 сек

Размер м: 128000, Проходов: 127998, Обменов: 127998, Время: 38.7771 сек

Среднее время для сортировки массива величиной 128000: 39.1554 сек



# Заключение.

Метод сортировки выбором очень прост и может быть использован в ситуации, когда нам нужно уменьшить количество замен. У него также существует несколько вариаций, которые можно использовать для разных ситуаций.