

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №5с

по дисциплине: Основы программирования

тема: «Оценка сложности алгоритмов сортировки по времени»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович

Проверили:
Притчин Иван Сергеевич
Черников Сергей Викторович

Белгород 2022 г.

Цель работы: получение навыков организации эксперимента и базового анализа результатов для оценки сложности алгоритмов по времени. Закрепление знаний о сортировках.

Содержание отчета:

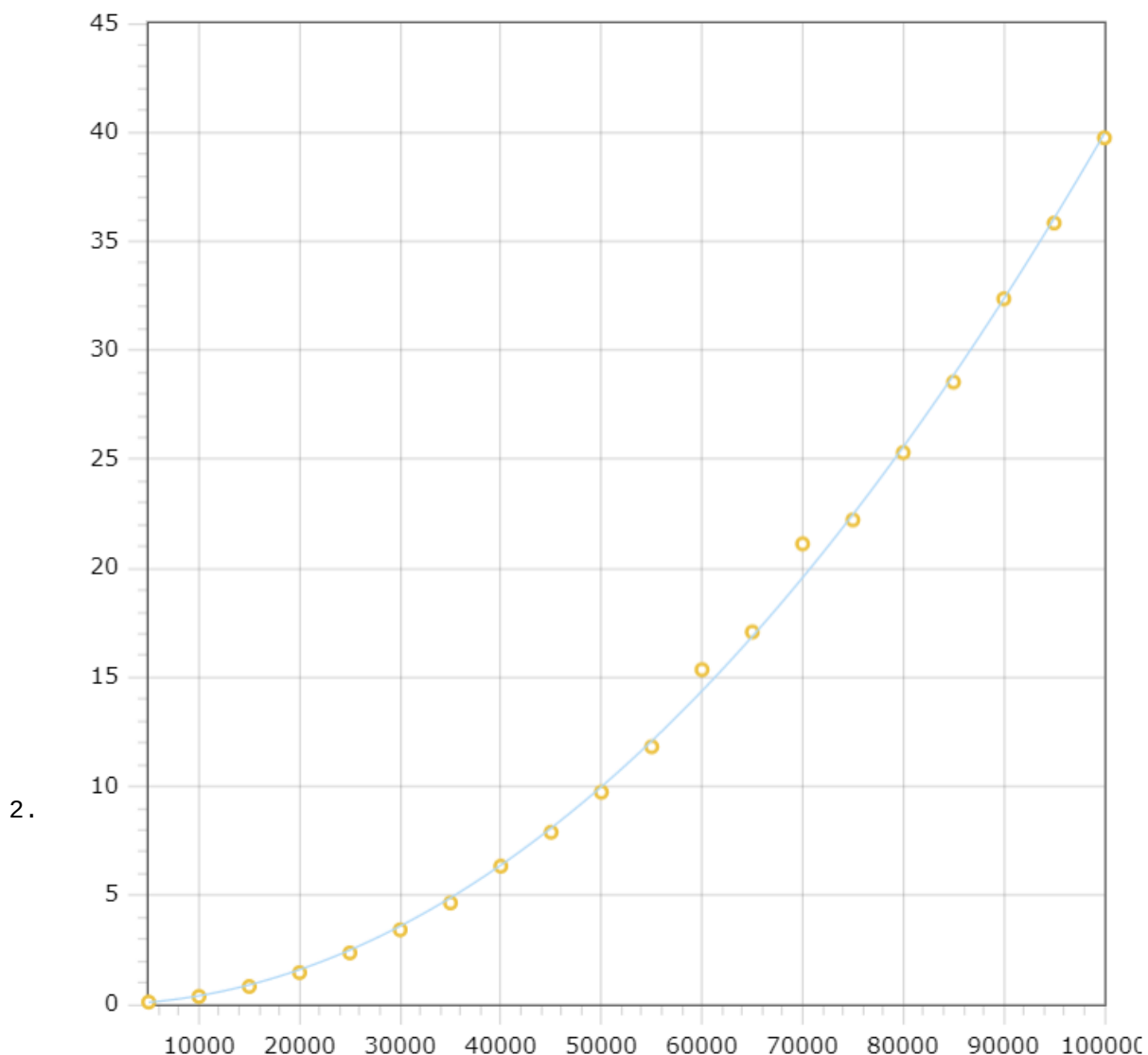
- Тема лабораторной работы.
- Цель лабораторной работы.
- Результаты замеров (таблицы по заданию)
- Графики полученных зависимостей.
- Исходный код экспериментов.
- Ссылка на открытый репозиторий с решением и полученными замерами и история коммитов (допускается текстом).

Ссылка на репозиторий: <https://github.com/ChuvilkoDEV/lab-5c>

Задания к лабораторной работе:

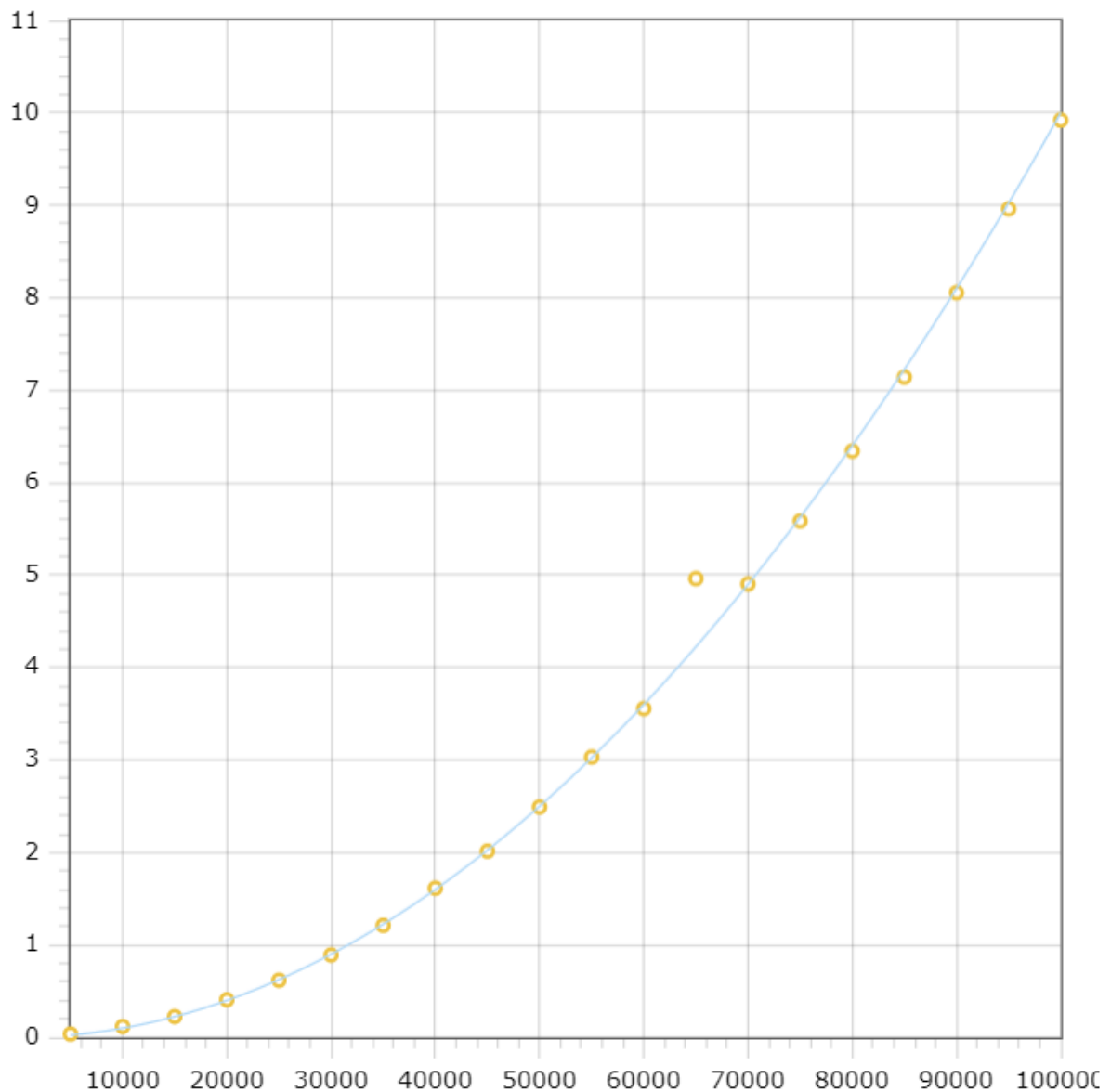
1. Пузырьковая сортировка

```
void bubbleSort(int *a, size_t n) {  
    for (int i = 0; i < n; ++i) {  
        for (int j = 0; j < n - 1; ++j)  
            if (a[j] > a[j + 1])  
                swap(&a[j], &a[j + 1]);  
    }  
}
```



Сортировка выбором

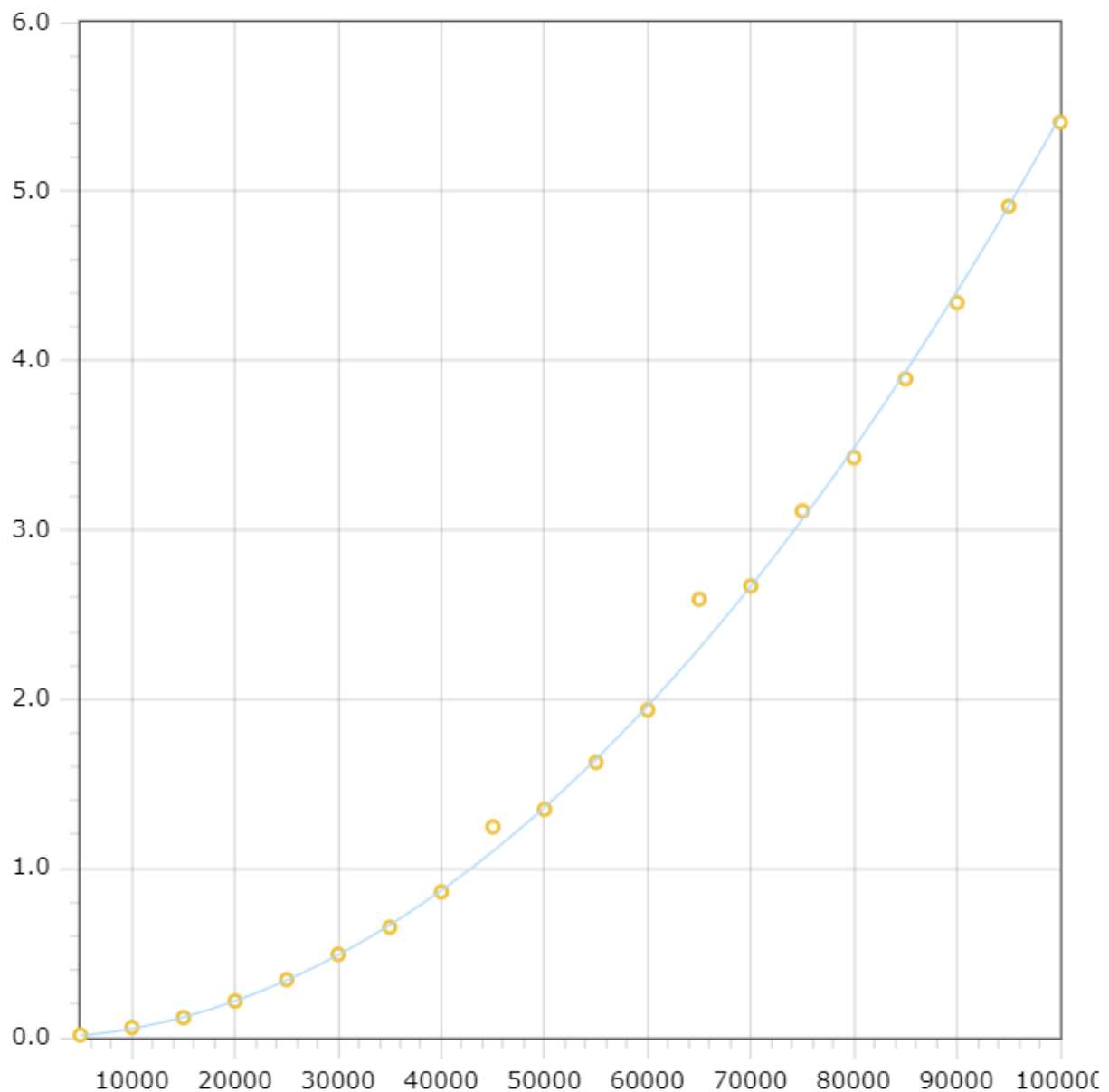
```
void selectionSort(int *a, size_t n) {  
    for (int i = 0; i < n; ++i) {  
        size_t minIndex = i;  
  
        for (int j = i + 1; j < n; ++j)  
            if (a[j] < a[minIndex])  
                minIndex = j;  
  
        swap(&a[i], &a[minIndex]);  
    }  
}
```



3. Сортировка вставками

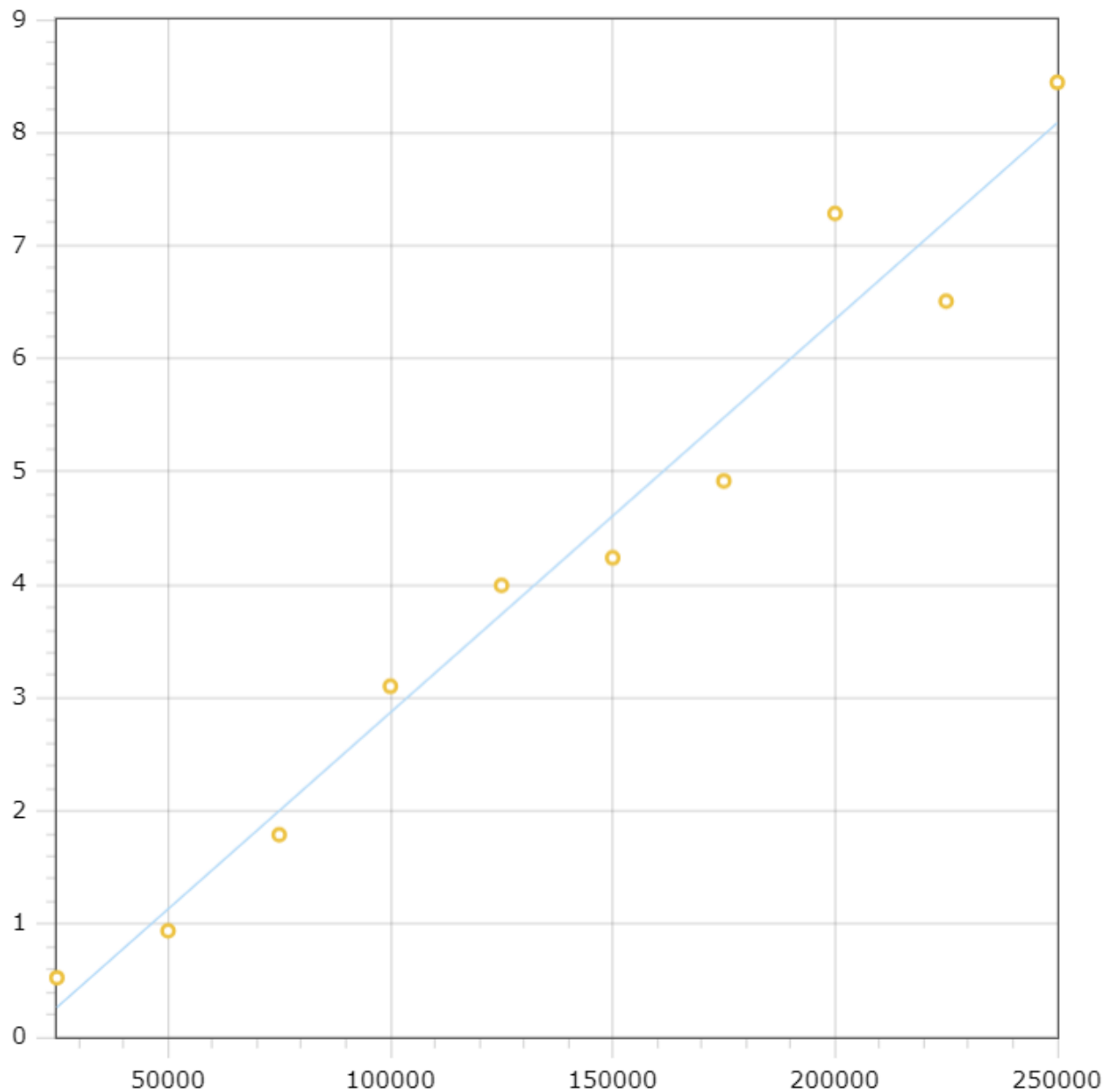
```
void insertionSort(int *a, size_t n) {  
    for (int i = 1; i < n; ++i) {  
        int t = a[i];  
        int j = i;  
        while (j > 0 && t < a[j - 1]) {  
            a[j] = a[j - 1];  
            j--;  
        }  
        a[j] = t;  
    }  
}
```

График на не отсортированном массиве



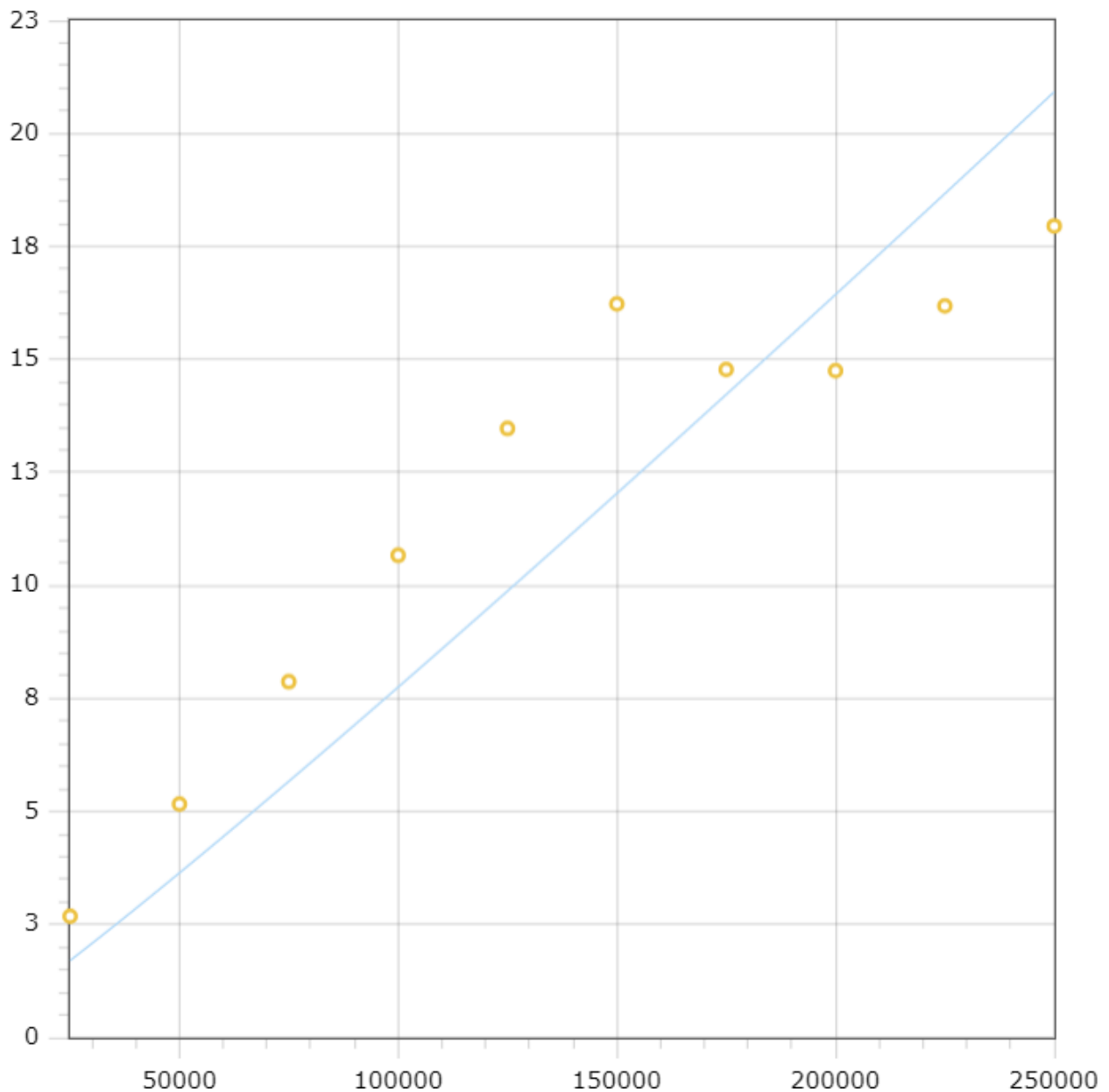
4. Сортировка расческой

```
void combsort(int *a, const size_t size) {
    size_t step = size;
    int swapped = 1;
    while (step > 1 || swapped) {
        if (step > 1)
            step /= 1.24733;
        swapped = 0;
        for (size_t i = 0, j = i + step; j < size; ++i, ++j)
            if (a[i] > a[j]) {
                swap(&a[i], &a[j]);
                swapped = 1;
            }
    }
}
```



5. Сортировка Шелла

```
void shellSort(int *a, size_t n) {  
    for (int i = n / 2; i > 0; i /= 2)  
        for (int j = i; j < n; ++j)  
            for (int k = j - i; k >= 0 && a[k] > a[k + i]; k -= i) {  
                int t = a[k];  
                a[k] = a[k + i];  
                a[k + i] = t;  
            }  
}
```



Ответы на вопросы:

1. Для того, чтобы сделать обменную сортировку, без использования операций обмена, можно присвоить каждому элементу свой индекс и производить операции над присвоенным числом индексом.

2. Результаты запуска:

```
-----  
Size : 200000  
Run #1|Name: selectionSort_random_time  
Status: OK! Time: 154.387
```

Предполагаемое время: 160 сек.

Фактическое время 154.387 сек.

Погрешность: 5.613 сек.

Таблица затраченного времени:

Затрачиваемое время сортировкой, чтобы упорядочить N элементов упорядоченного массива

Тип сортировки	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000	$T(N)$
Пузырьком	0.190	0.795	1.862	3.214	5.017	8.680	10.019	12.888	16.721	20.546	$2.0477379592198e-9 \cdot x^2$
Выбором	0.107	0.395	0.891	1.570	2.470	3.565	4.875	6.571	7.970	9.933	$1.00097273156681e-9 \cdot x^2$
Вставками	0.001	0.001	0.001	0.001	0.002	0.001	0.002	0.003	0.004	0.003	x

Тип сортировки	25000	50000	75000	100000	125000	150000	175000	200000	225000	250000	$T(N)$
Расческой (100 прогонов)	0.198	0.354	0.715	1.236	1.706	1.754	1.814	2.512	2.898	3.110	$1.33e-7 \cdot x - 0.2$
Шелла (1000 прогонов)	1.098	3.036	4.561	5.598	6.843	10.911	8.136	7.806	9.037	9.948	$3.77e-7 \cdot x \log x$

Затрачиваемое время сортировкой, чтобы упорядочить N элементов массива упорядоченного в обратном порядке

Тип сортировки	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000	$T(N)$
Пузырьком	0.318	1.243	2.807	4.992	7.803	14.095	15.334	19.939	26.078	31.526	$3.17080842332567e-9 \cdot x^2$
Выбором	0.111	0.375	0.863	1.516	2.357	3.400	4.764	6.077	7.682	9.551	$9.62681017285725e-10 \cdot x^2$
Вставками	0.105	0.432	0.962	1.719	2.682	3.876	5.317	6.898	8.997	10.821	$1.0905375374574e-9 \cdot x^2$

Тип сортировки	25000	50000	75000	100000	125000	150000	175000	200000	225000	250000	$T(N)$
Расческой (100 прогонов)	0.222	0.421	0.677	1.360	2.055	2.365	2.612	3.187	4.157	4.158	$1.89e-6 \cdot x - 0.483$
Шелла (1000 прогонов)	2.683	5.165	7.869	10.665	13.470	16.228	14.773	14.749	16.185	17.951	$6.73e-6 \cdot x \log x$

Затрачиваемое время каждой из сортировок для того, чтобы упорядочить N элементов неупорядоченного массива

Тип сортировки	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000	$T(N)$
Пузырьком	0.366	1.459	3.422	6.341	9.734	15.353	21.123	25.308	32.359	39.741	$3.99487148978398e-9 \cdot x^2$
Выбором	0.118	0.407	0.891	1.613	2.492	3.556	4.904	6.342	8.055	9.920	$1.00001057194967e-9 \cdot x^2$
Вставками	0.061	0.217	0.493	0.862	1.349	1.937	2.669	3.427	4.342	5.408	$5.44444376791216e-10 \cdot x^2$

Тип сортировки	25000	50000	75000	100000	125000	150000	175000	200000	225000	250000	$T(N)$
Расческой (100 прогонов)	0.526	0.940	1.789	3.102	3.996	4.238	4.919	7.283	6.508	8.443	$3.476e-7x - 0.6$
Шелла (1000 прогонов)	8.990	19.16	28.41	42.41	51.929	63.699	55.168	61.069	64.626	73.343	$2.67e-5x \log x$

Зависимость количества сравнений, производимых сортировкой от количества элементов в массиве (N)

Тип сортировки	Не упорядоченный	В обратном порядке	Упорядоченный	$\Phi BC(1)$	$\Phi BC(2)$	$\Phi BC(3)$
Пузырьком	$0.5N^2$	$0.99N^2$	$0.75N^2$	N^2	N^2	N^2
Выбором	$0.5N^2$	$0.53N^2$	$0.5N^2$	N^2	N^2	N^2
Вставками	$1.99N$	$0.99N^2$	$0.5N^2$	N	N^2	N^2
Расческой	$0.35N \log N$	$0.47N \log N$	$0.42N \log N$	$N \log N$	$N \log N$	$N \log N$
Шелла	$0.35N \log N$	$3.75N \log N$	$6.81N \log N$	$N \log N$	$N \log N$	$N \log N$