**Слайд1:** Добрый день. Тема моего доклад «Алгоритмы сортировок». В докладе речь пойдет про сортировку целочисленных элементов массива.

Проблема сортировки данных, является одной из наиболее актуальных и сложных для многих областей науки и технологий. Сортировка является одной из основных операций во многих приложениях и системах, и эффективные алгоритмы сортировки могут значительно повысить производительность программы. Каждый алгоритм сортировки обладает своими уникальными особенностями и преимуществами, которые могут быть ценными в разных ситуациях. Разработка алгоритмов не стоит на месте, а продолжает развиваться, и появляются новые, которые стремятся улучшить производительность, приспособиться к новым требованиям и решать специфические задачи.

Целью моего научного исследования было изучить основные алгоритмы сортировок, провести их анализ и попробовать модифицировать существующий алгоритм.

**Слайд2:** Все алгоритмы сортировок по способу работы можно свести к трём категориям: выбором; вставками; обмен. Наиболее известными алгоритмами являются: пузырьковая сортировка, сортировка выбором, сортировка вставками, шейкерная сортировка, сортировка Шелла, сортировка расческой, сортировка слиянием, быстрая сортировка, пирамидальная сортировка.

**Слайд3:** Был проведен их анализ. Определены скорости работы алгоритмов в нотации О-большое. Результаты исследований приведены в таблице.

**Слайд4:** Первой гипотезой ускорения работы алгоритма было усовершенствование сортировки выбором. Предполагалось, что проверка текущего элемента массива на максимальное и минимальное значение даст возможность за один проход найти эти значения и поставить по краям массива. Следующей итерацией выбирался подмассив без первого и последнего элемента. И так проходило до тех пор, пока границы подмассива не встретятся в середине исходного массива. Данному решению было дано название двусторонняя сортировка. Реализация представлена на слайде.

Однако при сравнении ее с представленными выше сортировками она не давала ожидаемых результатов. Но мы не стали останавливаться на этом.

**Слайд5:** Далее, при анализе алгоритмов была выдвинута еще гипотеза о возможности распараллеливания вычислений. Суть заключается в том, что массив, который мы хотим отсортировать, делится на две части. Далее, происходит параллельная сортировка этих частей и затем их объединение.

Была предпринята попытка распараллелить перечисленные выше сортировки. После анализа мы пришли к следующему выводу, что этот метод неплохо показывает себя при распараллеливании быстрой сортировки.

Часть кода, в которой происходит распараллеливание, представлен на слайде.

**Слайд6:** Были проведены эксперименты на массивах с разным количеством данных, по 100 итераций на каждый. На основе полученных результатов был построен график зависимости времени от количества чисел в массиве.

**Слайд7:** По результатам сравнения были сделаны следующие выводы:

1) данный алгоритм при сортировке массива, в котором количество элементов меньше 1000, не дает ускорения базовой версии быстрой сортировки

2) однако, данный алгоритм дает значительный выигрыш в скорости, если количество элементов в массиве превышает 1000 элементов.

Подводя итоги, можно сказать, что распараллеливание алгоритмов сортировки может быть эффективным при работе с большими объемами данных и может привести к значительному улучшению производительности. Однако, для небольших массивов, где накладные расходы на распараллеливание могут превышать выигрыш, базовые версии алгоритмов сортировки могут оставаться более оптимальным выбором.