

Bubble:

```
pts/activate
Enter N: 10000
Sequential time: 519 ms
Parallel time: 483 ms
PS C:\Users\Timing\Documents\GitHub\Heterogeneous-Parallelization\practices>
```

Selection:

```
Enter N: 10000
Sequential time: 182 ms
Parallel time: 181 ms
PS C:\Users\Timing\Documents\GitHub\Heterogeneous-Parallelization\practices>
```

Insertion:

```
Enter N: 10000
Sequential time: 128 ms
Parallel time: 130 ms
PS C:\Users\Timing\Documents\GitHub\Heterogeneous-Parallelization\practices>
```

```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

Enter N: 100000
Sequential time: 12978 ms
Parallel time: 13032 ms
PS C:\Users\Timing\Documents\GitHub\Heterogeneous-Parallelization\practices>
```

Контрольные вопросы

1. Отличия пузырька, выбора и вставок

Bubble sort постоянно меняет соседние элементы.

Selection sort ищет минимум и ставит его на место.

Insertion sort вставляет элементы в уже отсортированную часть массива.

2. Почему вставки сложно распараллелить?

Потому что каждый шаг зависит от результата предыдущего:

элементы сдвигаются и меняют позиции, из-за этого потоки мешают друг другу.

3. Какие директивы OpenMP использовались?

#pragma omp parallel for

`#pragma omp critical`

`reduction` (в других работах для сумм)

4. Плюсы и минусы параллельной сортировки

Плюсы: быстрее на больших массивах.

Минусы: накладные расходы, синхронизация, иногда медленнее последовательной.

5. Как измеряется производительность?

С помощью `std::chrono`, измеряя время до и после выполнения алгоритма.

6. Что происходит при увеличении числа потоков?

Сначала скорость растёт, потом упирается в память и синхронизацию и почти не увеличивается.

7. Когда параллельная сортировка хуже?

При маленьких массивах и при алгоритмах с сильными зависимостями (например, `insertion sort`).