Pawara Somsongkul 6280283

Functional & Parallel Programming

Project name: Parallel sorting

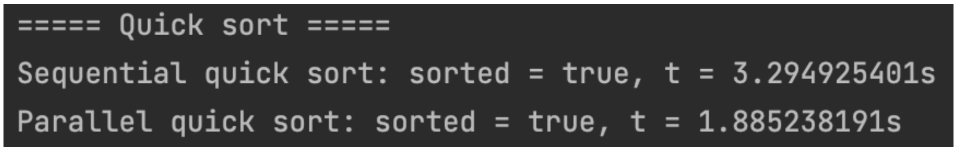
Language: Rust

Library used: Rayon, Crossbeam, Rand

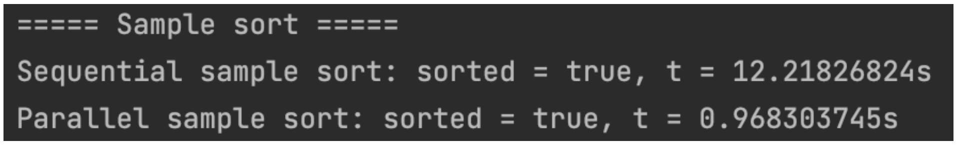
I implemented 3 algorithms of sorting: sample sort, Radix sort, and Quick sort. I have written both sequential and parallel versions (missing parallel radix sp). As we know, these three algorithms might be slow for a large scale of input. Therefore, we will parallelize them in order to minimize the running time and sort more data, amounts that can't be sorted serially.

- For quick sort, I used 1,000,000 random numbers in the vector
- For sample sort, I used 10,000 random numbers in the vector (I cannot use 1,000,000 numbers because it takes more than 15 minutes to complete compilation)
- For radix sort, I used 1,000,000 random numbers in the vector

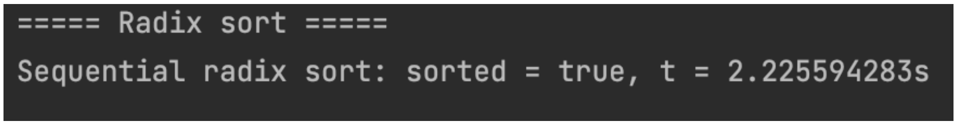
The result:



```
===== Quick sort =====
Sequential quick sort: sorted = true, t = 3.294925401s
Parallel quick sort: sorted = true, t = 1.885238191s
```



```
===== Sample sort =====
Sequential sample sort: sorted = true, t = 12.21826824s
Parallel sample sort: sorted = true, t = 0.968303745s
```



```
===== Radix sort =====
Sequential radix sort: sorted = true, t = 2.225594283s
```

Problem:

1) Type of input and output - I had to fix the code to make it able to sort both positive and negative numbers in the vector.
2) Understanding each sorting algorithm
3) Waiting for the infinity of Sample sort - I cannot use a random million number as another sorting.
4) Unfinished making parallel radix sorting :(