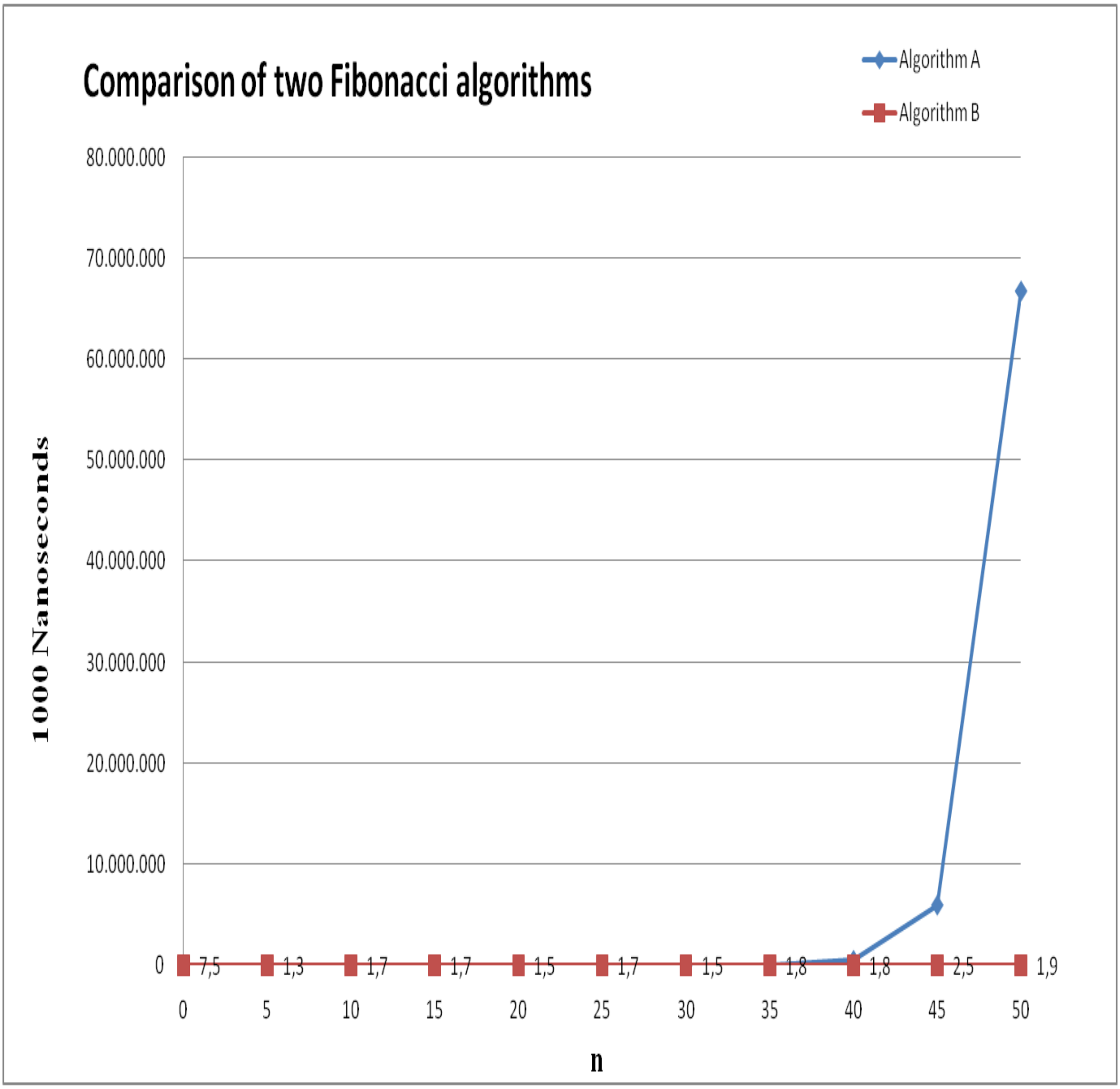
Tolgahan Arslan 22003061

Recursive method (calcFibA) output:

```
For n= 0 Fib_0 is calculated in 2500 nanoseconds.

For n= 5 Fib_5 is calculated in 1600 nanoseconds.

For n= 10 Fib_10 is calculated in 15400 nanoseconds.

For n= 15 Fib_15 is calculated in 87000 nanoseconds.

For n= 20 Fib_20 is calculated in 103800 nanoseconds.

For n= 25 Fib_25 is calculated in 775400 nanoseconds.

For n= 30 Fib_30 is calculated in 4371700 nanoseconds.

For n= 35 Fib_35 is calculated in 48700100 nanoseconds.

For n= 40 Fib_40 is calculated in 536127200 nanoseconds.

For n= 45 Fib_45 is calculated in 5997127900 nanoseconds.

For n= 50 Fib_50 is calculated in 66774281400 nanoseconds.
```

Non-recursive method (calcFibB) output:

```
For n= 0 Fib_0 is calculated in 7500 nanoseconds.

For n= 5 Fib_5 is calculated in 1300 nanoseconds.

For n= 10 Fib_10 is calculated in 1700 nanoseconds.

For n= 15 Fib_15 is calculated in 1700 nanoseconds.

For n= 20 Fib_20 is calculated in 1500 nanoseconds.

For n= 25 Fib_25 is calculated in 1700 nanoseconds.

For n= 30 Fib_30 is calculated in 1500 nanoseconds.

For n= 35 Fib_35 is calculated in 1800 nanoseconds.

For n= 40 Fib_40 is calculated in 1800 nanoseconds.

For n= 45 Fib_45 is calculated in 2500 nanoseconds.

For n= 50 Fib_50 is calculated in 1900 nanoseconds.
```



Recursive method takes much more time than non-recursive one because recursive functions needs to call simpler inputs in order to calculate the result. For this reason, it has to store more data in memory for proper calculation. Recursive functions work efficient with small numbers and data. However, when the numbers are large and there are huge amount of data (for example if we increase n from 50 to 10000) memory consumption will increase. This situation may cause problems since if the compiler runs out of memory; it will lead to a stack overflow exception (Benčević, 2018).

References

Benčević, M. (2018, August 31). *Why recursion is less efficient than a loop - programming word of the day*. Medium. Retrieved December 3, 2021, from https://medium.com/background-thread/why-recursion-is-less-efficient-than-a-loop-programming-word-of-the-day-514a5fdf3fc.