Avaliação de desempenho

Lab3: Bernardo Ventura 119164074

Caso 1.1: N = 10^5 com 1 thread

```
Valor mínimo sequencial: 7687602.000000
Valor máximo sequencial: 999994294272.000000
Valor mínimo concorrente: 7687602.000000
Valor máximo concorrente: 999994294272.000000
Tempo sequencial: 0.000502
Tempo concorrente: 0.000912
Aceleração: 0.550421
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$ ./lab3 100000 1
```

Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente foi maior que o sequencial, não valendo a pena o tempo de overhead para criação de uma única thread.

Caso 1.2: N = 10^5 com 2 threads

```
Valor mínimo sequencial: 5676876.500000
Valor máximo sequencial: 999998357504.000000
Valor mínimo concorrente: 5676876.500000
Valor máximo concorrente: 999998357504.000000
Tempo sequencial: 0.000505
Tempo concorrente: 0.000949
Aceleração: 0.532275
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$
```

Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente continuou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de duas threads.

Caso 1.3: N = 10^5 com 4 threads

```
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$ ./lab3 100000 4

Valor mínimo sequencial: 9118579.000000
Valor máximo sequencial: 999984005120.000000
Valor mínimo concorrente: 9118579.000000
Valor máximo concorrente: 999984005120.000000
Tempo sequencial: 0.000505
Tempo concorrente: 0.001565
Aceleração: 0.322775
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$
```

Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente continuou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de quatro threads.

Caso 2.1: N = 10^7 com 1 thread

```
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$ ./lab3 10000000 1]

Valor mínimo sequencial: 94529.242188
Valor máximo sequencial: 999999668224.000000
Valor mínimo concorrente: 94529.242188
Valor máximo concorrente: 999999668224.000000
Tempo sequencial: 0.023708
Tempo concorrente: 0.026568
Aceleração: 0.892332
```

Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de uma thread. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 1.1.

Caso 2.2: N = 10^7 com 2 threads

```
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$ ./lab3 10000000 2

Valor mínimo sequencial: 172294.671875
Valor máximo sequencial: 999999864832.000000
Valor mínimo concorrente: 172294.671875
Valor máximo concorrente: 999999864832.000000
Tempo sequencial: 0.022872
Tempo concorrente: 0.028959
Aceleração: 0.789808
```

Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de duas threads. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 1.2.

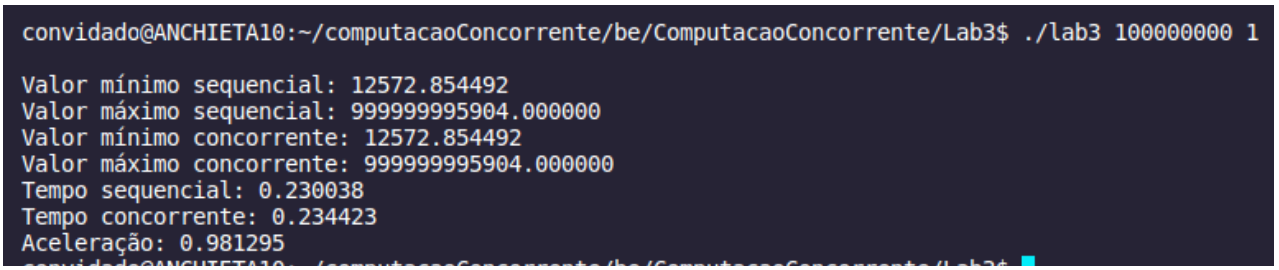
Caso 2.3: N = 10^7 com 4 threads

```
convidado@ANCHIETA10:~/computacaoConcorrente/be/ComputacaoConcorrente/Lab3$ ./lab3 10000000 4

Valor mínimo sequencial: 179745.250000
Valor máximo sequencial: 999999995904.000000
Valor mínimo concorrente: 179745.250000
Valor máximo concorrente: 999999995904.000000
Tempo sequencial: 0.022489
Tempo concorrente: 0.028420
Aceleração: 0.791329
```

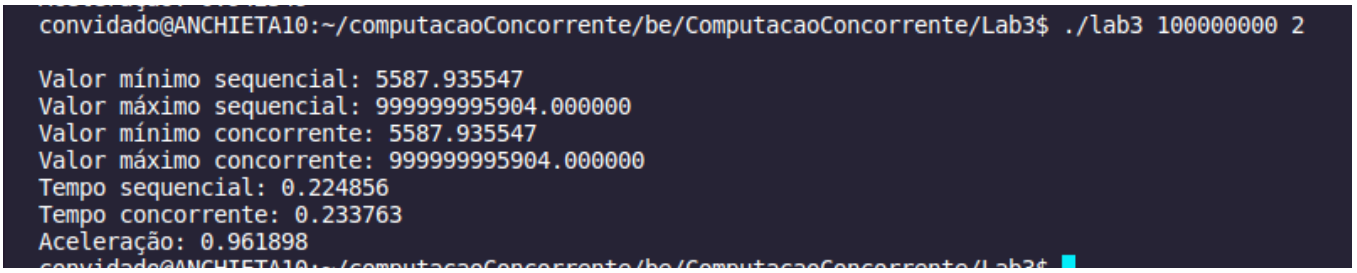
Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de quatro threads. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 1.3.

Caso 3.1: N = 10^8 com 1 thread.



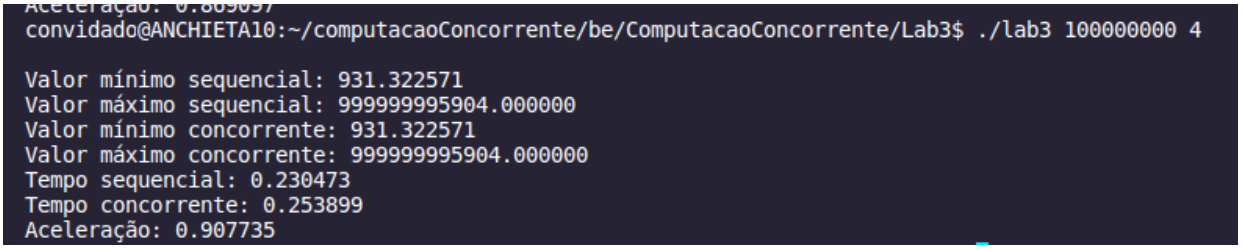
Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de uma thread. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 2.1. É possível observar que agora os tempos ficaram muito próximos.

Caso 3.2: N = 10^8 com 2 threads.



Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de duas threads. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 2.2. É possível observar que agora os tempos ficaram muito próximos.

Caso 3.3: N = 10^8 com 4 threads.



Realizei o teste 5 vezes, essa foi a que teve o melhor desempenho. Foi observado que o tempo concorrente ficou maior que o sequencial, então ainda não vale a pena o tempo de overhead para a criação de duas threads. Mas como o número do vetor aumentou, a aceleração ficou um pouco maior se comparado com o caso 3.2. É possível observar que agora os tempos ficaram muito próximos, porém o tempo concorrente com 4 threads foi o pior se comparado com 2 threads e com uma thread.