

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Programação Paralela e Distribuída

Relatório de Atividade Prática

Prof. Hélio

Alunos:

Caio Ueda Sampaio, 802215, BCC

Lucas Maciel Balieiro, 800534, BCC

Vinícius de Oliveira Guimarães, 802431, BCC

Gabriel Kusumota Nadalin, 819498, BCC

São Carlos/SP - 2024

1 - LINK PARA O COLAB COM OS CÓDIGOS E ANÁLISES

https://colab.research.google.com/drive/1FW7v166CSilz7y6huT0HqTcFXD_WvEae?usp=sharing

2 - TABELA DE TEMPOS DOS RESULTADOS

1000	Sequencial Dinâmico	OPENMP Dinâmico	CUDA Dinâmico	Sequencial Não Dinâmico	OPENMP Não Dinâmico	CUDA Não Dinâmico
real	0m0.007s	0m0.079s	0m0.175s	0m0.409s	0m0.201s	0m0.256s
user	0m0.007s	0m0.588s	0m0.020s	0m0.403s	0m0.649s	0m0.137s
sys	0m0.000s	0m0.000s	0m0.142s	0m0.004s	0m0.010s	0m0.113s

5000	Sequencial Dinâmico	OPENMP Dinâmico	CUDA Dinâmico	Sequencial Não Dinâmico	OPENMP Não Dinâmico	CUDA Não Dinâmico
real	0m0.188s	0m2.215s	0m0.181s	0m53.139s	0m21.974s	0m3.456s
user	0m0.186s	0m16.598s	0m0.043s	0m53.020s	1m11.197s	0m3.286s
sys	0m0.001s	0m0.040s	0m0.122s	0m0.028s	0m0.000s	0m0.108s

10000	Sequencial Dinâmico	OPENMP Dinâmico	CUDA Dinâmico	Sequencial Não Dinâmico	OPENMP Não Dinâmico	CUDA Nao Dinâmico
real	0m0.563s	0m8.210s	0m0.255s	7m16.812s	4m39.354s	0m24.891s
user	0m0.559s	1m0.298s	0m0.089s	7m14.220s	15m22.456s	0m24.572s
sys	0m0.000s	0m1.554s	0m0.153s	0m0.764s	0m0.070s	0m0.133s

3 - RESULTADOS EM GRÁFICOS

Gráfico 1 - Encontrar a subsequência de tamanho 3 em array com 1000 elementos

Array de 1000 elementos

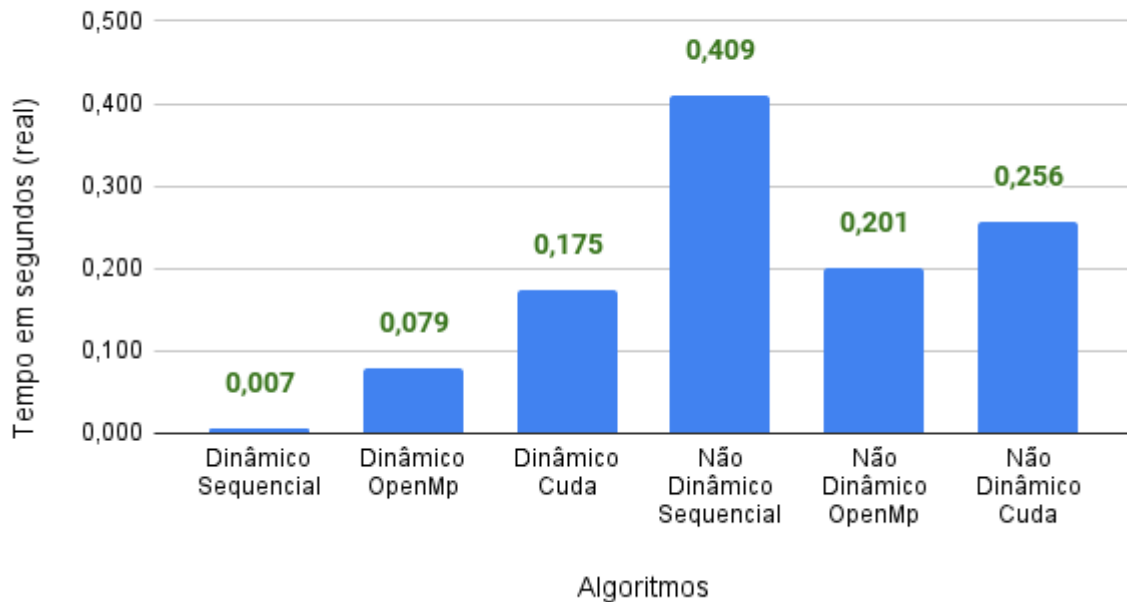


Gráfico 2 - Encontrar a subsequência de tamanho 3 em array com 5000 elementos

Array de 5000 elementos

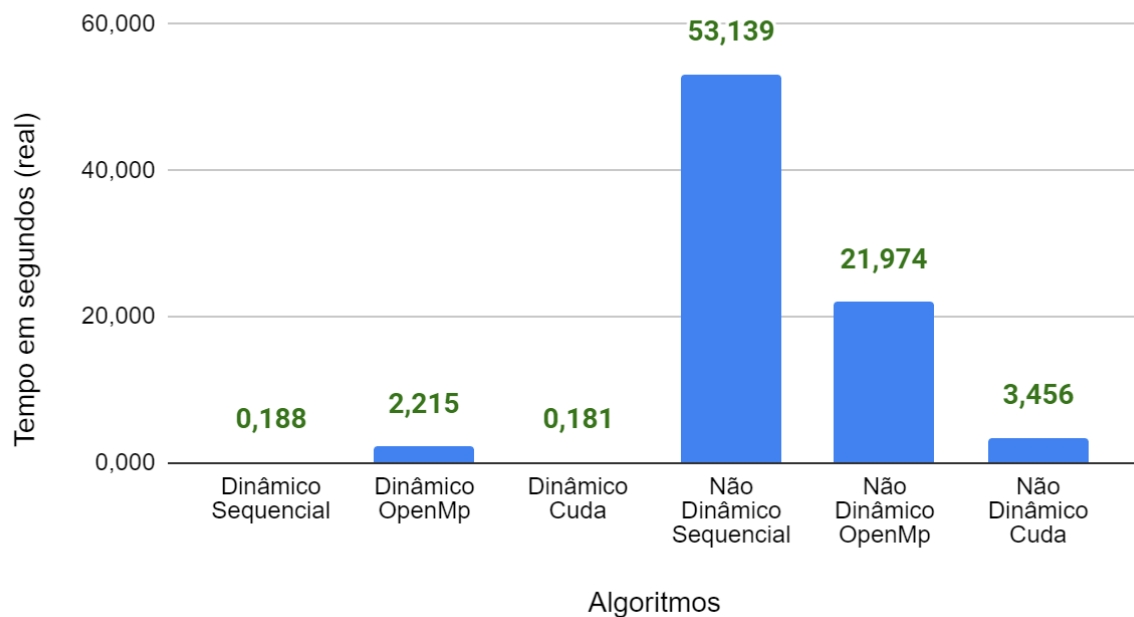
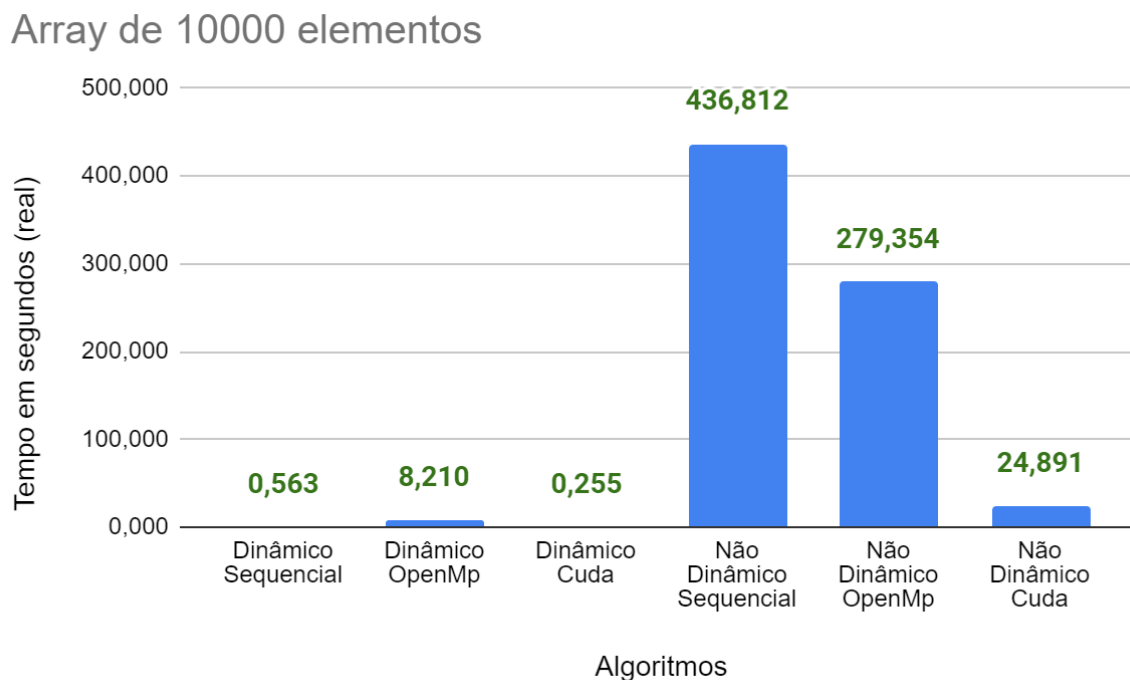


Gráfico 3 - Encontrar a subsequência de tamanho 3 em array com 10000 elementos



4 - CONCLUSÃO

Após a realização de diversos experimentos aplicados ao problema da subsequência de soma máxima, foi possível observar a grande vantagem da paralelização na busca de resultados mais rápidos.

Podemos concluir que programas com maior exigência computacional se beneficiam mais da paralelização quando comparado a algoritmos já feitos com mais otimização: Entre o não dinâmico sequencial e não dinâmico com Cuda, notamos um crescimento de 94% de eficiência, porém ao compararmos os mesmo em relação ao código dinâmico, notamos um aumento de apenas 54%, e a paralelização por OpenMP do código dinâmico foi inclusive detrimental para a eficiência de tempo.

Vale apontar que isso se deve, em parte, ao fato de que o código dinâmico já era extremamente eficiente, o que é demonstrado pelo fato de que a execução mais lenta do código dinâmico ainda foi mais rápida do que a melhor execução de código não dinâmico.