

Universidade de São Paulo  
Instituto de Matemática e Estatística  
IME

## MiniEP7 - Programação Concorrente e Paralela

Patrícia da Silva Rodrigues (n<sup>o</sup>USP 11315590)

Junho  
2023

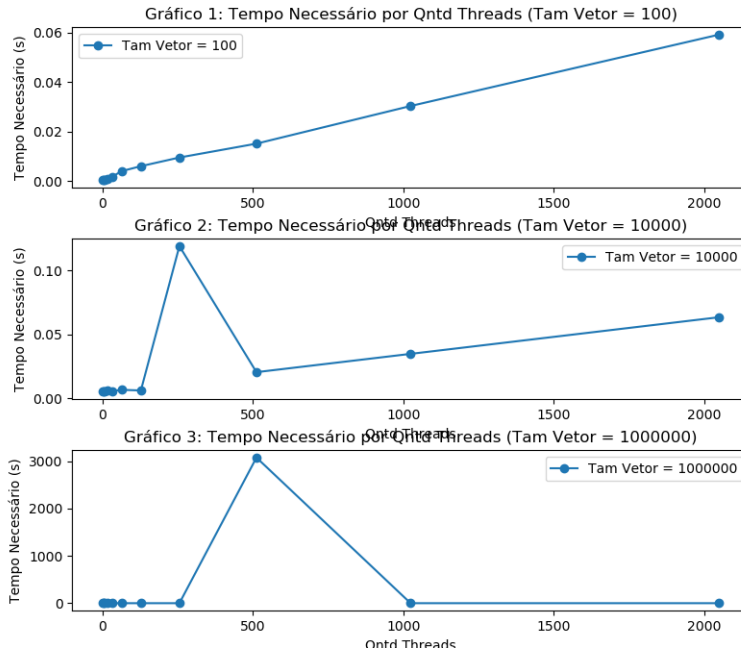
3 de junho de 2023

# 1 Testando diferentes tamanhos de vetores e quantidades de threads

Responde o pedido 1:

Tabela 1: Tabela de resultados			
Qntd	Threads	Tam Vetor	Tempo Necessário (s)
	2	100	0.0004
	4	100	0.0006
	8	100	0.0007
	16	100	0.0010
	32	100	0.0018
	64	100	0.0041
	128	100	0.0061
	256	100	0.0096
	512	100	0.0152
	1024	100	0.0304
	2048	100	0.0592
	2	10000	0.0052
	4	10000	0.0054
	8	10000	0.0058
	16	10000	0.0060
	32	10000	0.0057
	64	10000	0.0067
	128	10000	0.0062
	256	10000	0.0119
	512	10000	0.0205
	1024	10000	0.0348
	2048	10000	0.0634
	2	1000000	0.3388
	4	1000000	0.3000
	8	1000000	0.3016
	16	1000000	0.3008
	32	1000000	0.2989
	64	1000000	0.2995
	128	1000000	0.2996
	256	1000000	0.3022
	512	1000000	0.3076
	1024	1000000	0.3214
	2048	1000000	0.3534

## 2 Graficos



## 3 Análise

Responde a parte 2:

Podemos observar pela tabela e gráfico que:

Quando a quantidade de threads aumenta: Para o vetor de tamanho 100, o tempo necessário diminui inicialmente à medida que o número de threads aumenta de 2 para 8. No entanto, a partir de 16 threads, o tempo começa a aumentar novamente. Para o vetor de tamanho 10000, o tempo necessário diminui consistentemente à medida que o número de threads aumenta. Para o vetor de tamanho 1000000, o tempo necessário inicialmente diminui com o aumento das threads, mas estabiliza após um determinado ponto.

Quando o tamanho do vetor aumenta: Para 2 threads, o tempo necessário para executar a tarefa aumenta significativamente à medida que o tamanho do vetor aumenta. No entanto, para 4 ou mais threads, o tempo necessário para executar a tarefa permanece relativamente constante à medida que o tamanho do vetor aumenta.

Além disso, o impacto do tamanho do vetor no tempo de execução pode variar dependendo do número de threads. Para um número menor de threads, um vetor maior pode resultar em tempos de execução mais longos, enquanto

para um número maior de threads, o tamanho do vetor pode ter um efeito limitado no tempo de execução.

Responde o pedido 3:

O if dentro da seção crítica previne que corra uma condição de corrida (race condition). Caso tivéssemos duas ou mais threads que tenham passado pelo if externo, verificando que a condição é verdadeira. Se não houvesse um if interno dentro da seção crítica, todas as threads tentariam atualizar a variável max simultaneamente, causando resultados incorretos e muitos problemas.