

## Questão 1 D

letra A

	$10^5$	tempo	$10^6$	tempo	$10^7$	tempo	$10^8$	tempo	$10^{12}$	tempo
8 thread	3.14158	0.00826	3.141583	0.004814	3.1415827	0.0060290	3.14158265	0.00100800	3.141582653590	0.00100800
16 threads	3.14158	0.00383	3.141583	0.002915	3.1415827	0.0025780	3.14158265	0.00102500	3.141582653590	0.001631000

letra B

	$10^5$	tempo	$10^6$	tempo	$10^7$	tempo	$10^8$	tempo	$10^{12}$	tempo
8 thread	3.14158	0.00120	3.141583	0.002028	3.1415827	0.0006580	3.14158265	0.00127200	3.141582653590	0.00127200
16 threads	3.14158	3.14158	3.141583	3.141583	3.1415827	0.0014060	3.14158265	0.00163100	3.141582653590	0.001072000

### Tempo de Execução:

#### 8 Threads:

- A tabela A apresenta tempos de execução consistentemente mais rápidos do que a tabela B. A diferença é especialmente significativa para cálculos de  $10^5$  e  $10^6$  iterações, com a tabela A sendo mais rápida.
- A tabela B apresenta um tempo de execução anormalmente alto para o cálculo de  $10^7$  iterações, sendo mais lento que a tabela A.

#### 16 Threads:

- A tabela A novamente apresenta tempos de execução mais rápidos para a maioria dos casos, com exceção do cálculo de  $10^7$  iterações, onde a diferença é mínima.
- A diferença de tempo entre as tabelas B com 8 e 16 threads é insignificante para todos os casos, indicando que a escalabilidade com 16 threads é ineficaz.

### Valores de PI:

- Ambas as tabelas geram os mesmos valores de PI para todas as configurações de threads e iterações, confirmando a precisão dos cálculos.

### Conclusão:

A tabela A demonstra um desempenho superior em comparação à tabela B, com tempos de execução consistentemente mais rápidos para a maioria dos casos.

Recomendações:

### Questão 2

número de threads	m=8000000, n=8 T médio(10 execuções)	dobro do número de threads	m=8000, n=8000 T médio(10 execuções)	dobro do número de threads	m=8, n=8000000 T médio(10 execuções)	dobro do número de threads
1	0,130622	0,261244	0,0190921	0,381842	0,0508695	0,101739
8	0,1564532	0,3129064	0,1172237	0,2344474	0,1310215 1	0,2620430 2

### Análise:

#### Dobrando o número de threads:

Reduziu o tempo de execução em 49% (m=8000000, n=8).

Reduziu o tempo de execução em 22% (m=8, n=8000000).

Aumento (5%) no tempo de execução (m=8000, n=8000).

**Conclusão:** O aumento do número de threads geralmente acelera a execução, mas o efeito pode ser menor em alguns casos.

#### Aumentando o número de termos:

Aumentou o tempo de execução em 2000% (m=8000, n=8).

Aumento (1%) no tempo de execução (m=8000000, n=8).

**Conclusão:** O aumento do número de termos aumenta significativamente o tempo de execução.

**Aumentando o Tamanho da Matriz:**

**Conclusão:** Aumenta significativamente o tempo de execução.