



Departamento de Ciência da Computação

# COMPLEXIDADE E EFICIÊNCIA EM ORDENAÇÃO:

## O ALGORITMO QUICKSORT COMO SOLUÇÃO PARA LIMITAÇÕES DE MÉTODOS QUADRÁTICOS

ARTHUR OLIVEIRA, ELIÁZARO FERREIRA, GIOVANNA PEREIRA

# CONTEXTUALIZAÇÃO

- O grande volume de dados exige o desenvolvimento de algoritmos capazes de processar conjuntos de dados cada vez maiores com eficiência.

# OBJETIVO

- O objetivo do artigo é apresentar um novo método para solucionar o problema de ordenação baseada na estratégia de divisão e conquista.

# TRABALHOS RELACIONADOS

## BUBBLE SORT

O mais ineficiente dos algoritmos. Faz muitas comparações e muitas trocas para ordenar o conjunto.

$O(N^2)$

## SELECTION SORT

Minimiza o número de trocas ao, em cada passagem, encontrar o menor elemento na lista não ordenada e trocá-lo com o primeiro.

$O(N^2)$

## INSERTION SORT

Faz uma quantidade menor de trocas ao colocar o elemento na posição correta dentro do vetor ordenado.

$O(N^2)$   
 $O(N)$

Melhor caso  
(vetores ordenados)

# METODOLOGIA

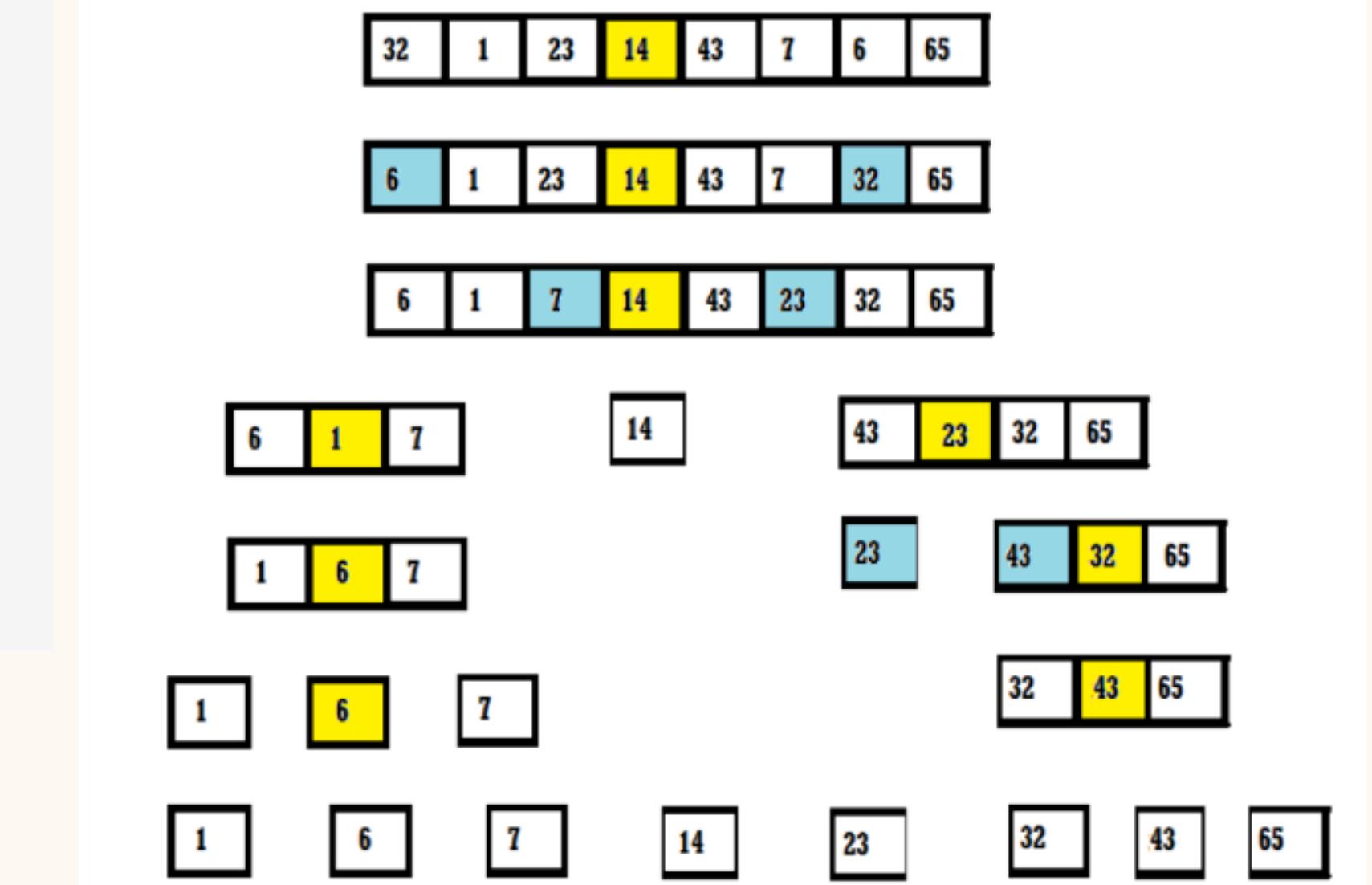
```
função particao(vetor, inicio, fim):
    pivo = vetor[(inicio + fim) div 2] // div para divisão inteira
    i = inicio
    j = fim

    enquanto i <= j:
        enquanto vetor[i] < pivo:
            i = i + 1
        enquanto vetor[j] > pivo:
            j = j - 1
        se i <= j:
            troca(vetor[i], vetor[j])
            i = i + 1
            j = j - 1

    retorna i

procedimento quicksort(vetor, inicio, fim):
    se inicio < fim:
        p = particao(vetor, inicio, fim)
        quicksort(vetor, inicio, p - 1)
        quicksort(vetor, p, fim)
```

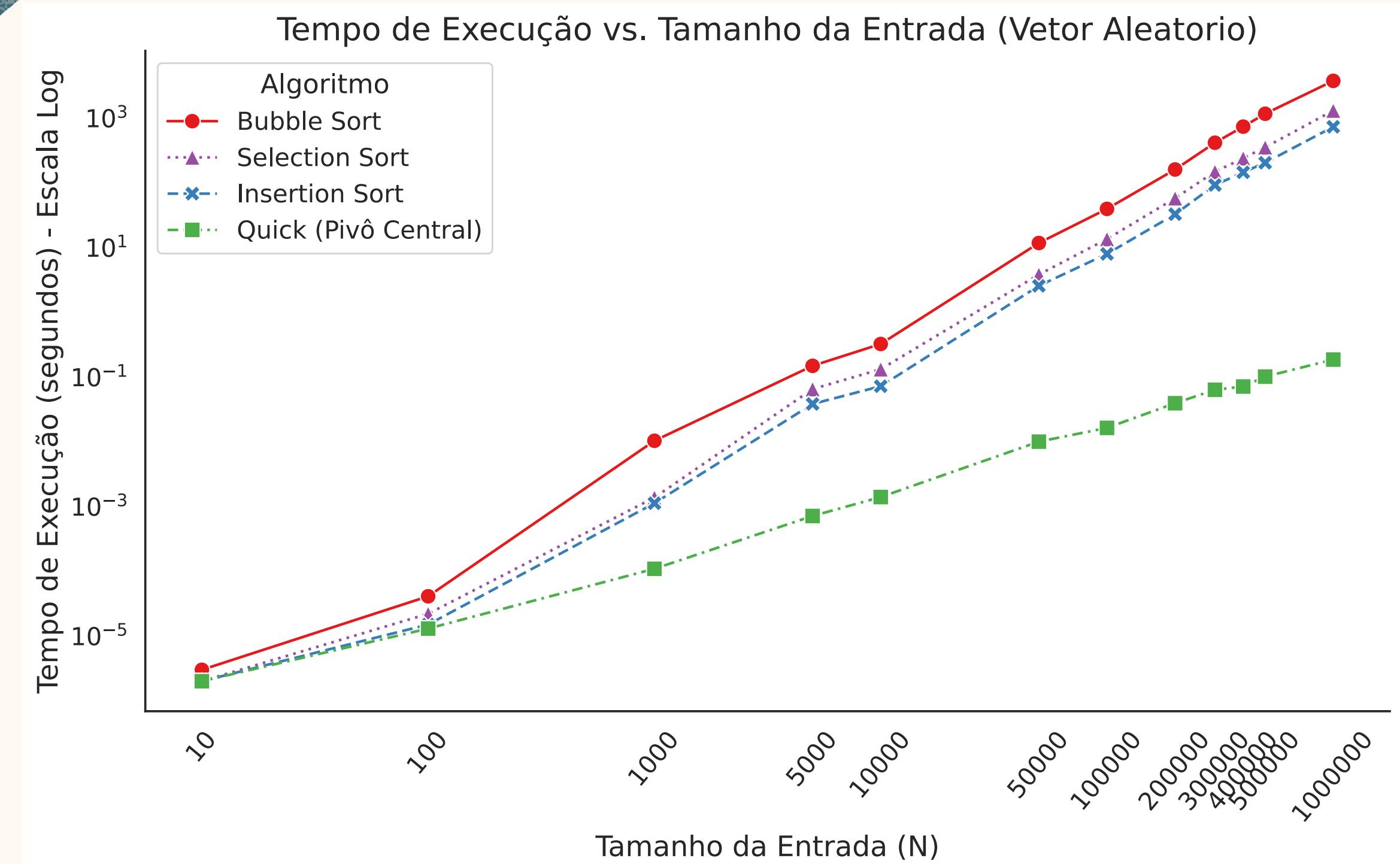
exemplo



$$T(n) = 2T(n/2) + O(n)$$

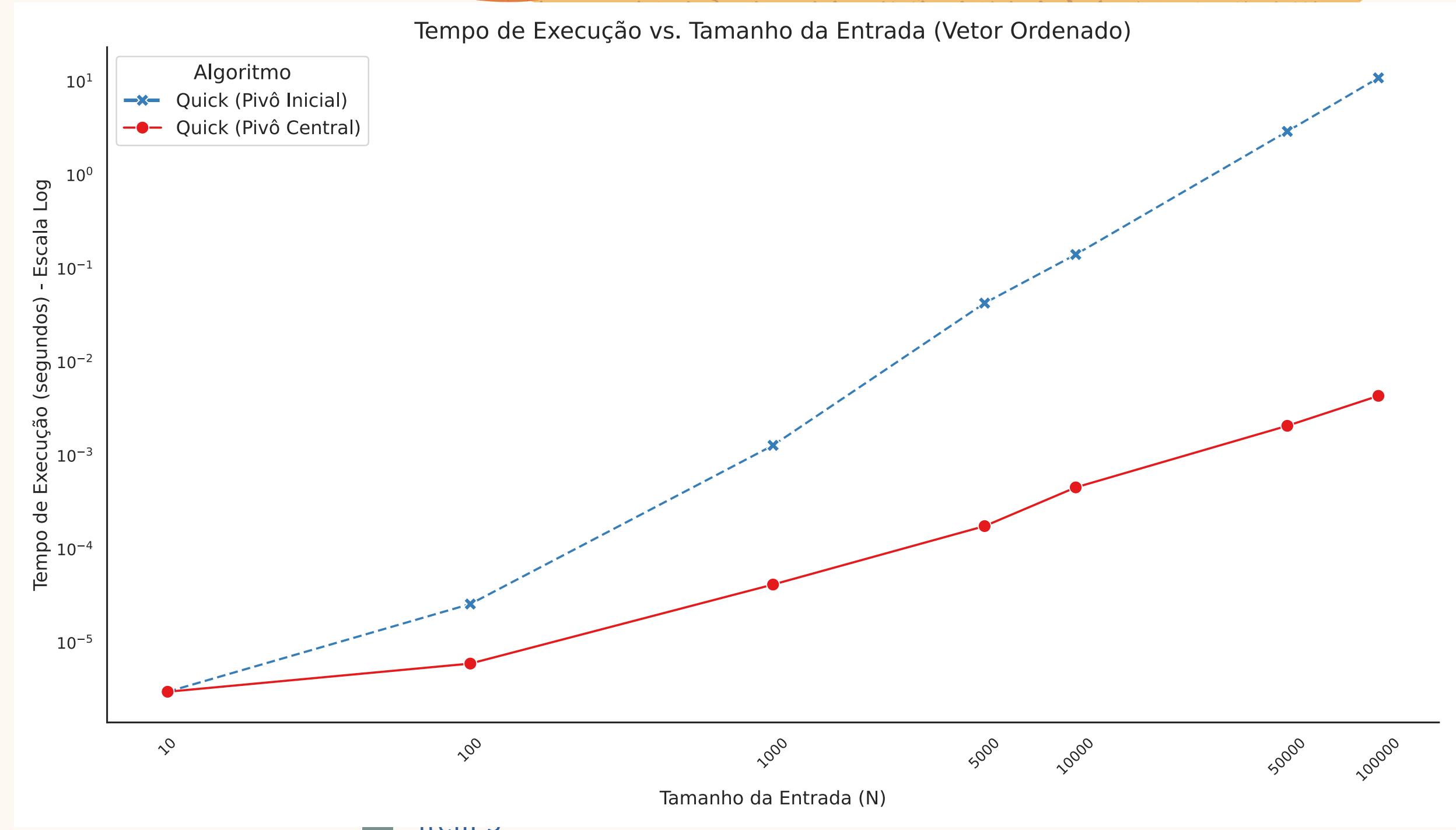
# RESULTADOS

Caso médio.



# RESULTADOS

Pior caso.



# LIMITAÇÕES

- Não é estável.
- A eficiência do algoritmo é altamente dependente da escolha do pivô.

# TRABALHOS FUTUROS

**Trabalhar em heurísticas para otimizar a escolha do pivô visando aprimorar seu desempenho, bem como em formas de tornar o algoritmo estável.**

# REFERÊNCIAS

- A. Autor. Análise comparativa entre algoritmos de ordenação. Livro B, Sem Local, 1970.
- T. Someone, C. E. Somebody, and R. L. Everyone. Bubble sort: The simplest sorting algorithm ever made. ACM Journal of sorting algorithms (ACM JSA), 2(2):155–165, 1950.
- T. Someone, C. E. Somebody, and R. L. Everyone. Selection sort: A better algorithm. In IEEE Conference on sorting algorithms (IEEE CSA), Sem Local, 1960.
- T. Someone, C. E. Somebody, and R. L. Everyone. Insertion sort: State-of-the-art sorting algorithm. In ACM Conference of sorting algorithms for computer science (ACM SACS), Sem Local, 1965.
- Lucimara C. FERREIRA, Rickson C. RODRIGUES, Eric C. RAPOSO, Xayanne H. de SOUZA, Amanda M. de OLIVEIRA, and Genarde M. TRINDADE. Pensamento computacional no 1º ano do ensino fundamental: Explorando o bubble sort por meio da dança. In WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 33., pages 585–594, Porto Alegre, 2025. Sociedade Brasileira de Computação.