



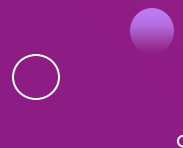
COMPARAÇÃO ENTRE ALGORITMOS: INSERTION SORT E QUICK SORT

Arthur Sant'anna
Bruno Almeida de Oliveira
Daniel Ojeda
Gabriel Theophilo
Rodrigo Machado





INTRODUÇÃO AO PROBLEMA DE ORDENAÇÃO



- A ordenação é um problema fundamental em ciência da computação e matemática, que envolve a organização de um conjunto de elementos em uma determinada ordem. O objetivo da ordenação é tornar os dados mais acessíveis e úteis para processamento posterior.
- Existem diversos algoritmos de ordenação, cada um com suas características e complexidades. Alguns dos algoritmos mais conhecidos são o Quicksort, o Mergesort, o Heapsort e o Bubblesort. Esses algoritmos diferem em eficiência, facilidade de implementação e uso de recursos computacionais.





APLICAÇÕES DO PROBLEMA



- **Bancos de Dados:** Em bancos de dados, a ordenação é frequentemente utilizada para recuperar dados de forma ordenada, facilitando a busca e a apresentação dos resultados.
- **Algoritmos de Busca:** Muitos algoritmos de busca, como a busca binária, requerem que os dados estejam ordenados para funcionarem corretamente e serem eficientes.
- **Aplicações Gráficas:** Em aplicações gráficas, a ordenação é frequentemente usada para ordenar elementos visuais, como gráficos ou listas, de acordo com critérios específicos.
- **Processamento de Texto e Dados:** A ordenação é útil em processamento de texto e dados, onde é necessário classificar informações alfabeticamente, numericamente ou por outros critérios.
- **Computação Paralela:** Em computação paralela, a ordenação é um problema desafiador devido à necessidade de coordenar a ordenação de múltiplos conjuntos de dados ao mesmo tempo.





INSERTION SORT

Insertion Sort é um algoritmo de ordenação simples que constrói a lista ordenada por um elemento por vez;

Melhor Caso ($O(n)$): Ocorre quando a lista já está ordenada. O algoritmo só precisa comparar cada elemento com o anterior, resultando em uma complexidade linear.

Caso Médio ($O(n^2)$): Para uma lista com elementos em ordem aleatória, o número de comparações e trocas ainda tende a ser quadrático.

Pior Caso ($O(n^2)$): Ocorre quando a lista está ordenada em ordem inversa. O algoritmo precisa comparar cada elemento com todos os anteriores, resultando em uma complexidade quadrática.

PSEUDOCÓDIGO

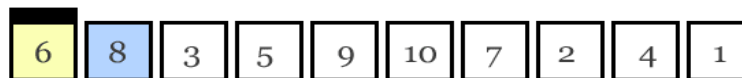
For $i : 1$ to $\text{lenght}(A) - 1$

$J = i$

 While $j > 0$ and $A[j-1] > A[j]$

 Swap $A[j]$ and $A[j-1]$

$J = j - 1$



Yellow is smallest number found

Blue is current item

Green is sorted list





QUICK SORT

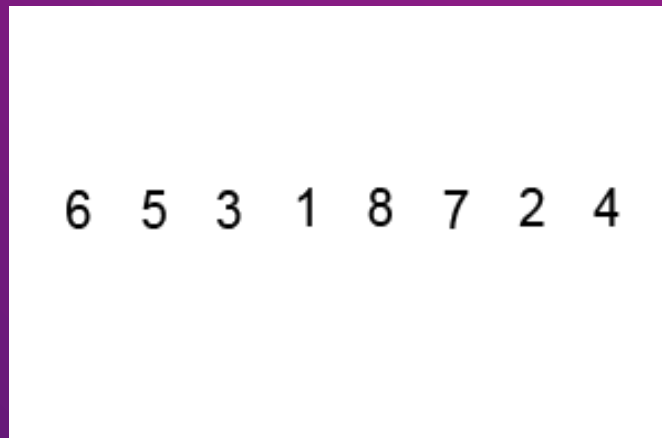
O Quicksort é o algoritmo mais eficiente na ordenação por comparação. Nele se escolhe um elemento chamado de pivô, a partir disto é organizada a lista para que todos os números anteriores a ele sejam menores que ele, e todos os números posteriores a ele sejam maiores que ele. Ao final desse processo o número pivô já está em sua posição final. Os dois grupos desordenados recursivamente sofrem o mesmo processo até que a lista esteja ordenada.

Melhor Caso ($O(n \log n)$): Ocorre quando o pivô sempre divide a lista em duas partes quase iguais.

Caso Médio ($O(n \log n)$):

Para listas com elementos em ordem aleatória, a probabilidade de uma partição balanceada é alta, resultando em uma complexidade esperada logarítmica.

Pior Caso ($O(n^2)$): Ocorre quando o pivô é sempre o menor ou maior elemento da lista, resultando em partições altamente desbalanceadas.





QUICK SORT PSEUDOCÓDIGO

```
function quicksort(arr, low, high):  
  if low < high:  
    pivot_index = partition(arr, low, high)  
    quicksort(arr, low, pivot_index - 1)  
    quicksort(arr, pivot_index + 1, high)
```

```
function partition(arr, low, high):  
  pivot = arr[high]  
  i = low - 1  
  for j from low to high - 1:  
    if arr[j] < pivot:  
      i = i + 1  
      swap arr[i] with arr[j]  
  swap arr[i + 1] with arr[high]  
  return i + 1
```





DESCRIÇÃO DAS INSTÂNCIAS

1

LISTAS PEQUENAS

- 10 Listas geradas aleatoriamente.
- 10 Listas ordenadas.
- 10 Listas ordenadas inversamente.
- As listas pequenas possuem entre 10 a 100 elementos.

2

LISTAS MÉDIAS

- 10 listas geradas aleatoriamente
- 10 Listas ordenadas.
- 10 Listas ordenadas inversamente.
- As listas médias possuem entre 950 a 5000 elementos.

3

LISTAS GRANDES

- 10 listas geradas aleatoriamente
- 10 Listas ordenadas.
- 10 Listas ordenadas inversamente.
- As listas grandes possuem entre 10.000 a 100.000





DESCRIÇÃO DO COMPUTADOR

NOTEBOOK ACER NITRO 5

Sistema Operacional: Windows 11

Placa de vídeo: GTX1650 4GB

Processador: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, 2400 Mhz, 4 Núcleo(s), 8 Processador(es) Lógico(s)

Memória: 16 Gb Ram





INTRODUÇÃO

APLICAÇÕES

INSERTION

QUICK

DESCRIÇÕES

ANÁLISE

CONCLUSÃO

ESTRUTURA DE DADOS

1

LISTAS PEQUENAS



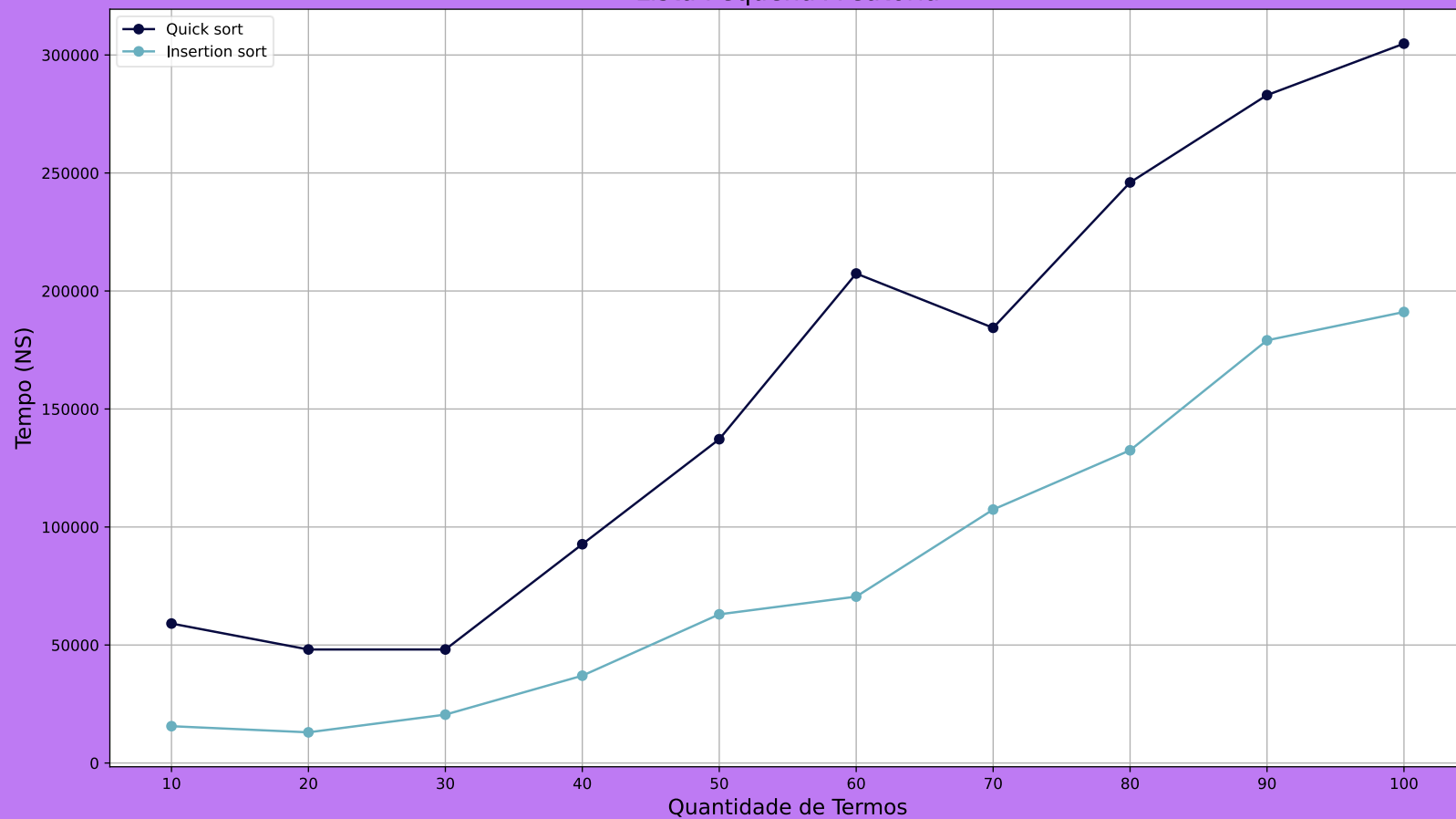
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Aleatoria	15600
20	Pequena Aleatoria	13000
30	Pequena Aleatoria	20500
40	Pequena Aleatoria	37000
50	Pequena Aleatoria	63000
60	Pequena Aleatoria	70500
70	Pequena Aleatoria	107400
80	Pequena Aleatoria	132500
90	Pequena Aleatoria	179100
100	Pequena Aleatoria	191100

QUICK SORT

Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Aleatoria	59100
20	Pequena Aleatoria	48100
30	Pequena Aleatoria	48100
40	Pequena Aleatoria	92700
50	Pequena Aleatoria	137200
60	Pequena Aleatoria	207400
70	Pequena Aleatoria	184400
80	Pequena Aleatoria	246000
90	Pequena Aleatoria	283000
100	Pequena Aleatoria	304800

Lista Pequena Aleatória





INSERTION SORT

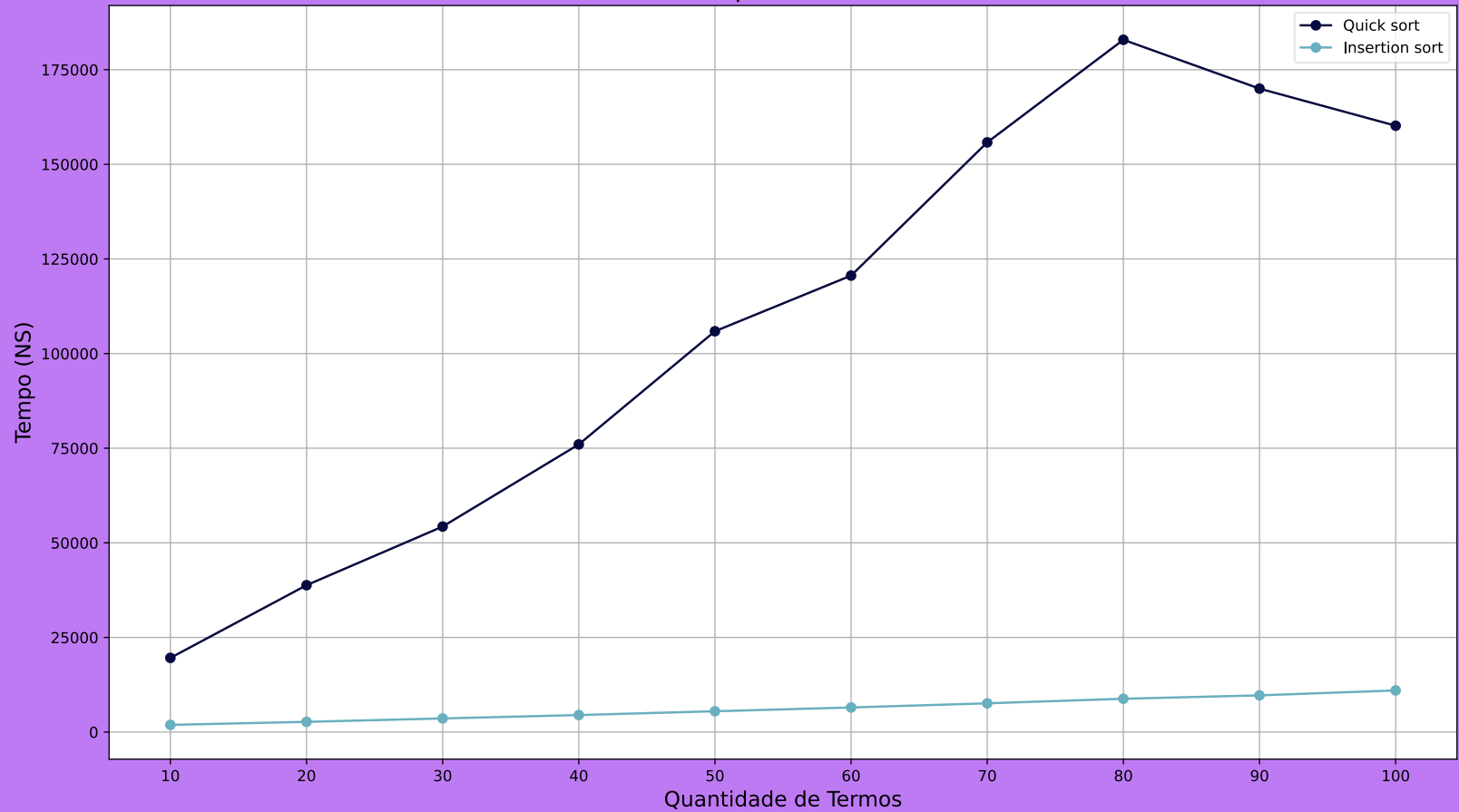
Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Ordenada	1900
20	Pequena Ordenada	2700
30	Pequena Ordenada	3600
40	Pequena Ordenada	4500
50	Pequena Ordenada	5500
60	Pequena Ordenada	6500
70	Pequena Ordenada	7600
80	Pequena Ordenada	8800
90	Pequena Ordenada	9700
100	Pequena Ordenada	11000



QUICK SORT

Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Ordenada	19600
20	Pequena Ordenada	38800
30	Pequena Ordenada	54300
40	Pequena Ordenada	76000
50	Pequena Ordenada	105900
60	Pequena Ordenada	120600
70	Pequena Ordenada	155800
80	Pequena Ordenada	182900
90	Pequena Ordenada	170000
100	Pequena Ordenada	160200

Lista Pequena Ordenada



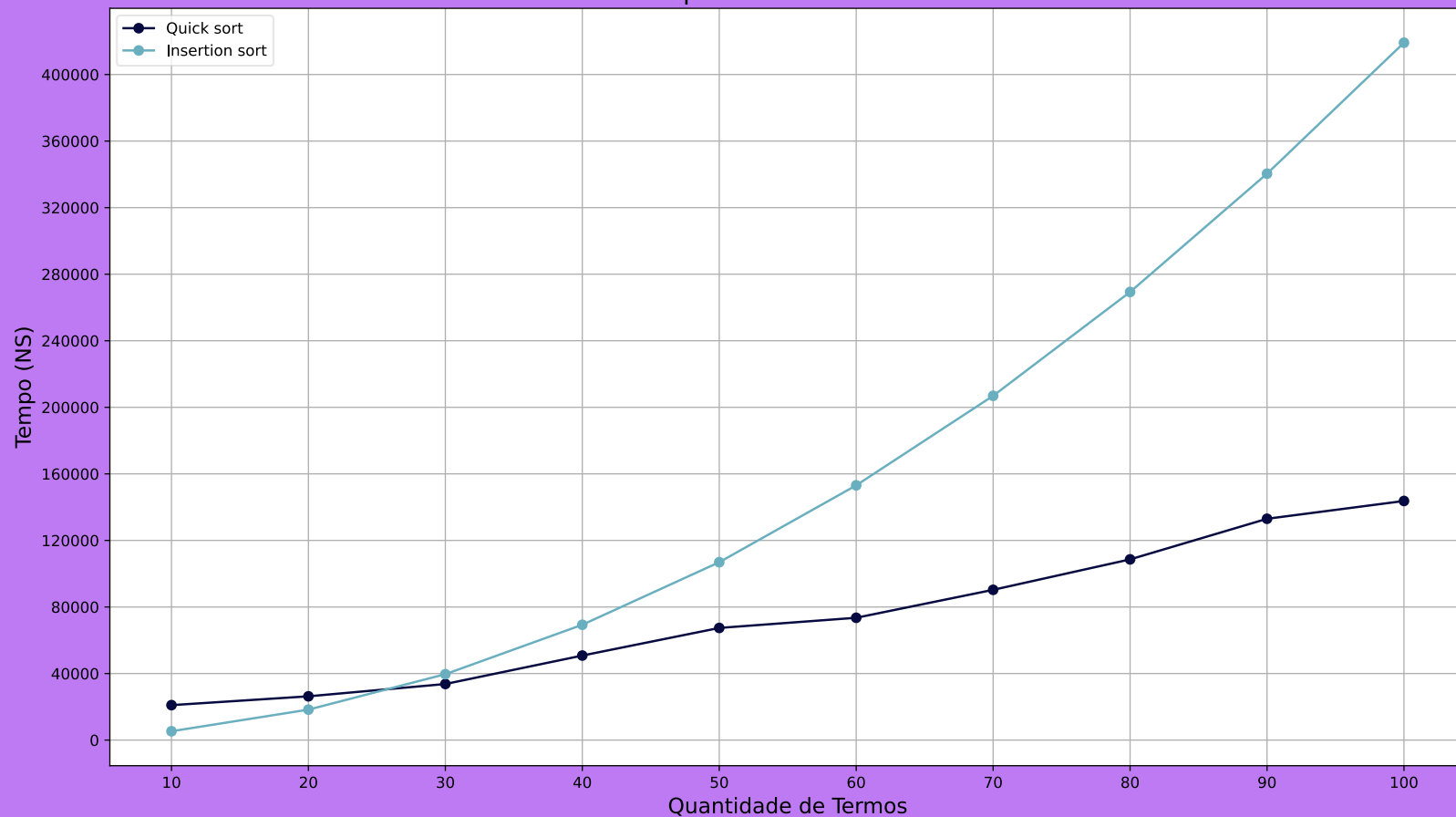
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Ordenada Reversa	5300
20	Pequena Ordenada Reversa	18300
30	Pequena Ordenada Reversa	39600
40	Pequena Ordenada Reversa	69300
50	Pequena Ordenada Reversa	106900
60	Pequena Ordenada Reversa	153100
70	Pequena Ordenada Reversa	206900
80	Pequena Ordenada Reversa	269300
90	Pequena Ordenada Reversa	340400
100	Pequena Ordenada Reversa	419100

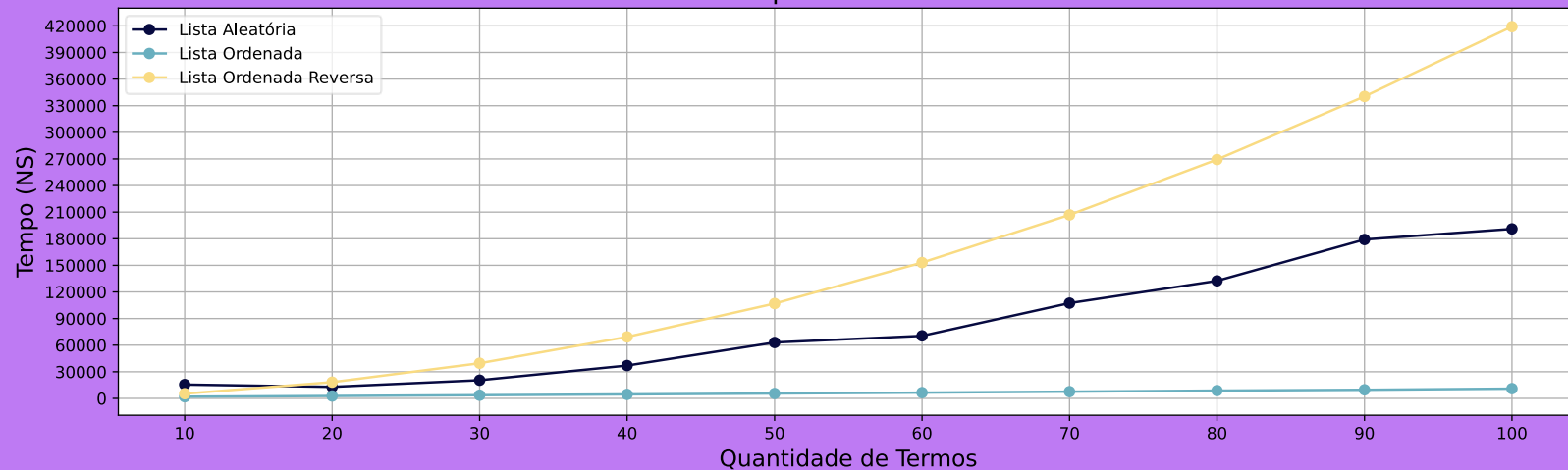
QUICK SORT

Tamanho	Tipo de Lista	Tempo (ns)
10	Pequena Ordenada Reversa	21000
20	Pequena Ordenada Reversa	26300
30	Pequena Ordenada Reversa	33700
40	Pequena Ordenada Reversa	50800
50	Pequena Ordenada Reversa	67400
60	Pequena Ordenada Reversa	73500
70	Pequena Ordenada Reversa	90300
80	Pequena Ordenada Reversa	108600
90	Pequena Ordenada Reversa	133000
100	Pequena Ordenada Reversa	143700

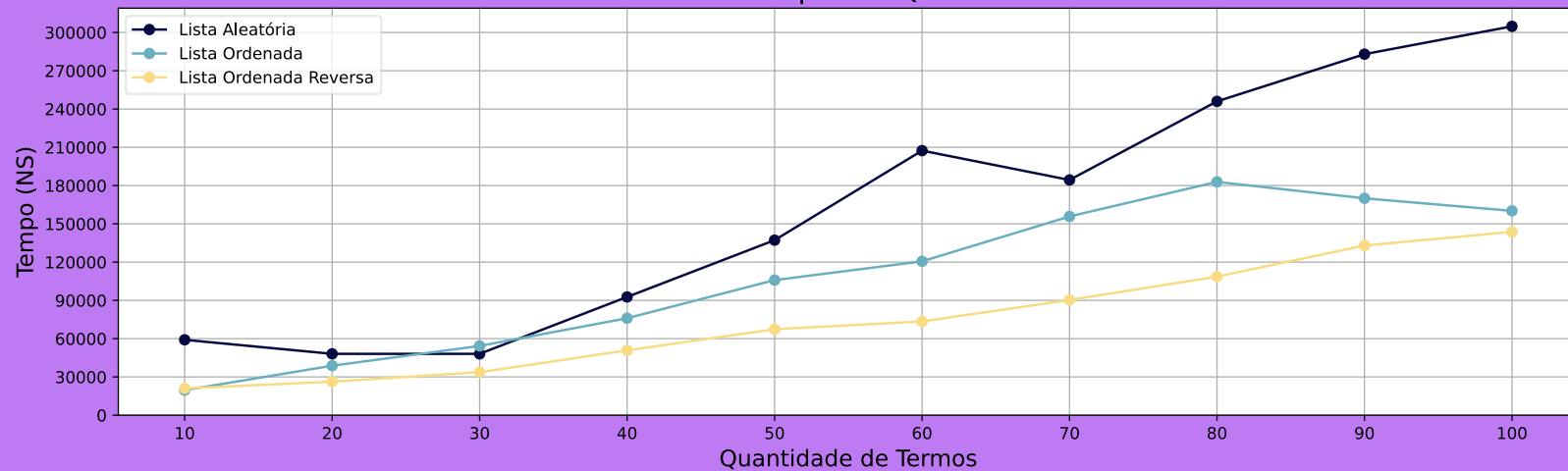
Lista Pequena Ordenada Reversa



Listas Pequenas Insertion sort



Listas Pequenas Quicksort





RESULTADOS(ALGORITMO MAIS RÁPIDO)



INSTÂNCIA	INSERTION	QUICK
Pequena Aleatória	X	
Pequena Ordenada	X	
Pequena Ordenada Reversa		X





INTRODUÇÃO

APLICAÇÕES

INSERTION

QUICK

DESCRIÇÕES

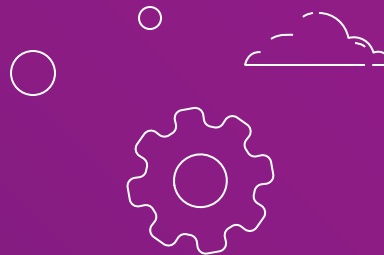
ANÁLISE

CONCLUSÃO

ESTRUTURA DE DADOS

2

LISTAS MÉDIAS



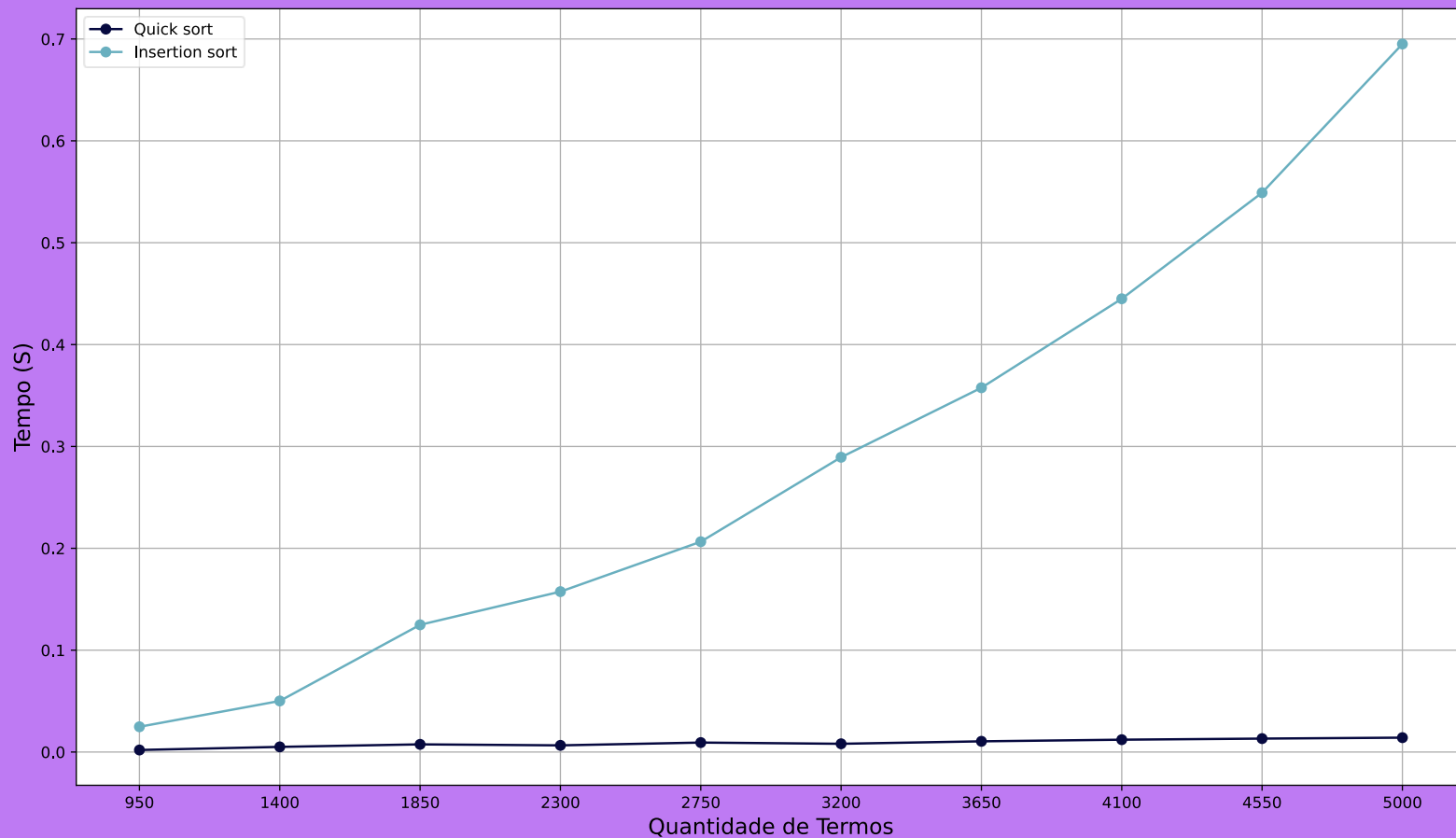
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (s)
950	Media Aleatoria	0,024789
1400	Media Aleatoria	0,050143
1850	Media Aleatoria	0,124883
2300	Media Aleatoria	0,157458
2750	Media Aleatoria	0,206475
3200	Media Aleatoria	0,289374
3650	Media Aleatoria	0,357631
4100	Media Aleatoria	0,444959
4550	Media Aleatoria	0,549075
5000	Media Aleatoria	0,69505

QUICK SORT

Tamanho	Tipo de Lista	Tempo (s)
950	Media Aleatoria	0,002006292
1400	Media Aleatoria	0,00503993
1850	Media Aleatoria	0,007525682
2300	Media Aleatoria	0,006542921
2750	Media Aleatoria	0,009285927
3200	Media Aleatoria	0,00809145
3650	Media Aleatoria	0,010499954
4100	Media Aleatoria	0,012116194
4550	Media Aleatoria	0,013180971
5000	Media Aleatoria	0,014104843

Lista Média Aleatoria



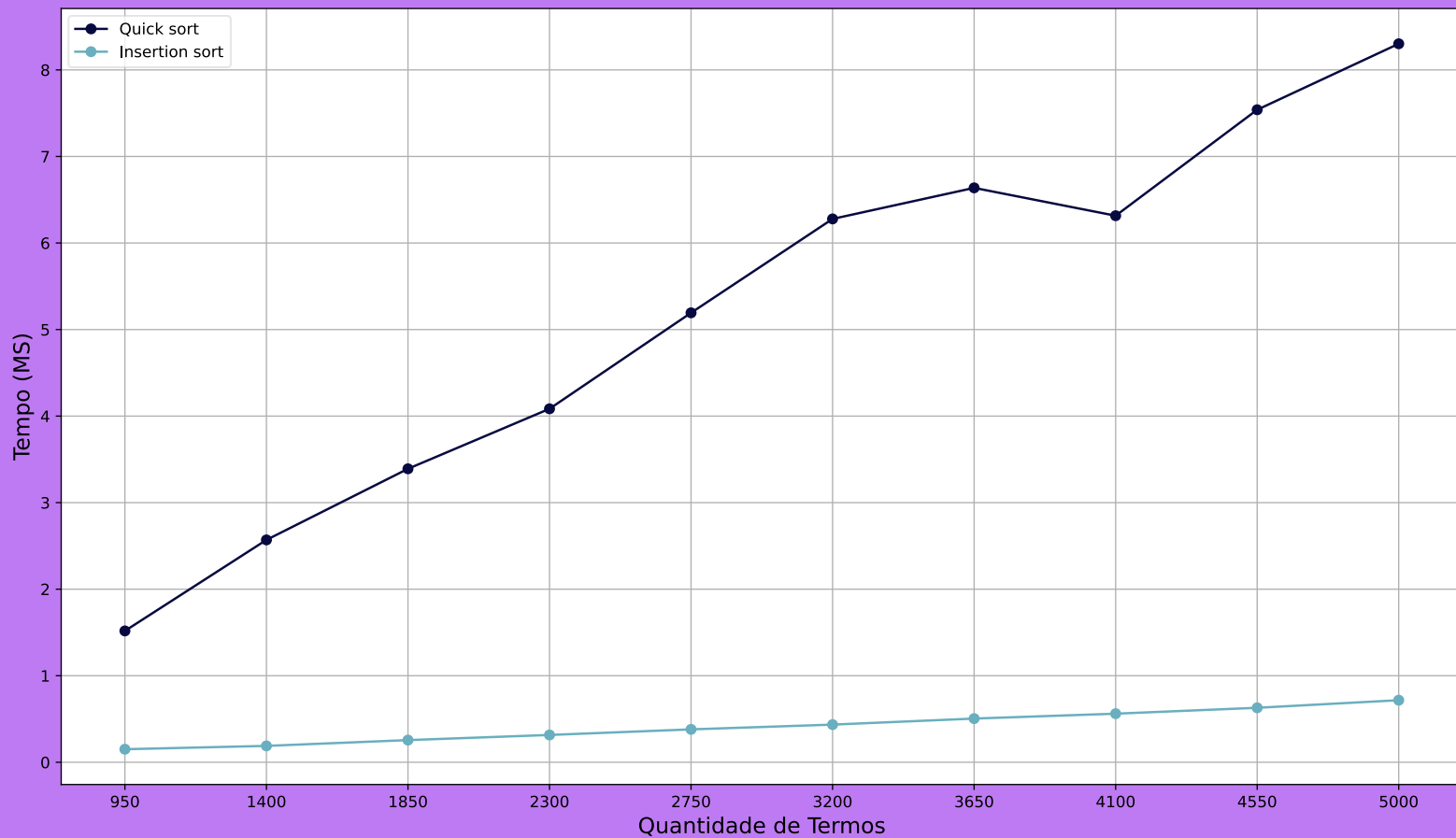
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (ns)
950	Media Ordenada	150700
1400	Media Ordenada	189200
1850	Media Ordenada	256100
2300	Media Ordenada	315600
2750	Media Ordenada	380100
3200	Media Ordenada	435100
3650	Media Ordenada	504800
4100	Media Ordenada	561300
4550	Media Ordenada	629300
5000	Media Ordenada	717200

QUICK SORT

Tamanho	Tipo de Lista	Tempo (ns)
950	Media Ordenada	1517600
1400	Media Ordenada	2569300
1850	Media Ordenada	3391000
2300	Media Ordenada	4084200
2750	Media Ordenada	5193200
3200	Media Ordenada	6278200
3650	Media Ordenada	6637100
4100	Media Ordenada	6315200
4550	Media Ordenada	7539600
5000	Media Ordenada	8301900

Lista Média Ordenada



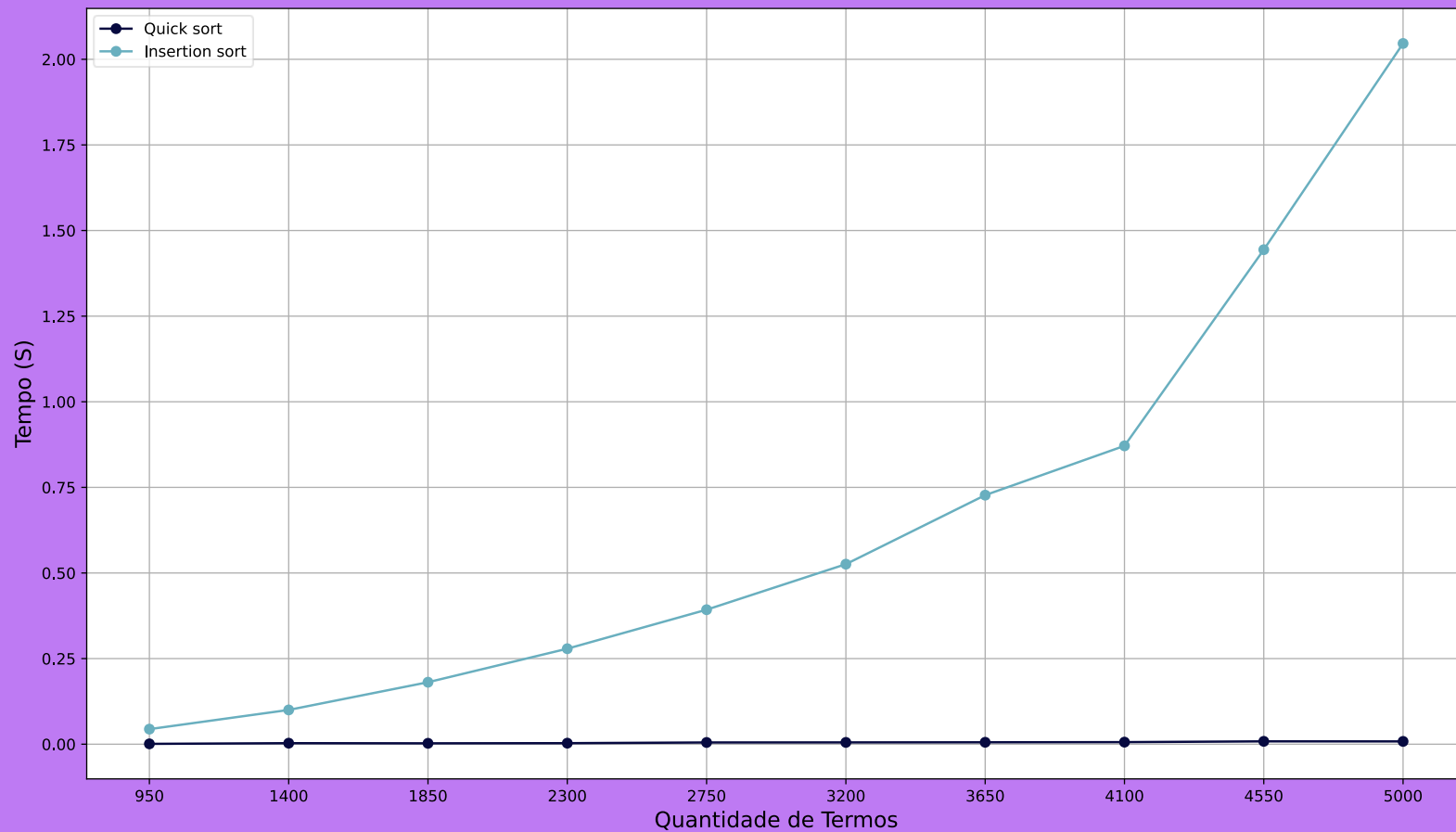
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (s)
950	Media Ordenada Reversa	0,043998
1400	Media Ordenada Reversa	0,100074
1850	Media Ordenada Reversa	0,180986
2300	Media Ordenada Reversa	0,278863
2750	Media Ordenada Reversa	0,392764
3200	Media Ordenada Reversa	0,525703
3650	Media Ordenada Reversa	0,727294
4100	Media Ordenada Reversa	0,871252
4550	Media Ordenada Reversa	1,444023
5000	Media Ordenada Reversa	2,046268

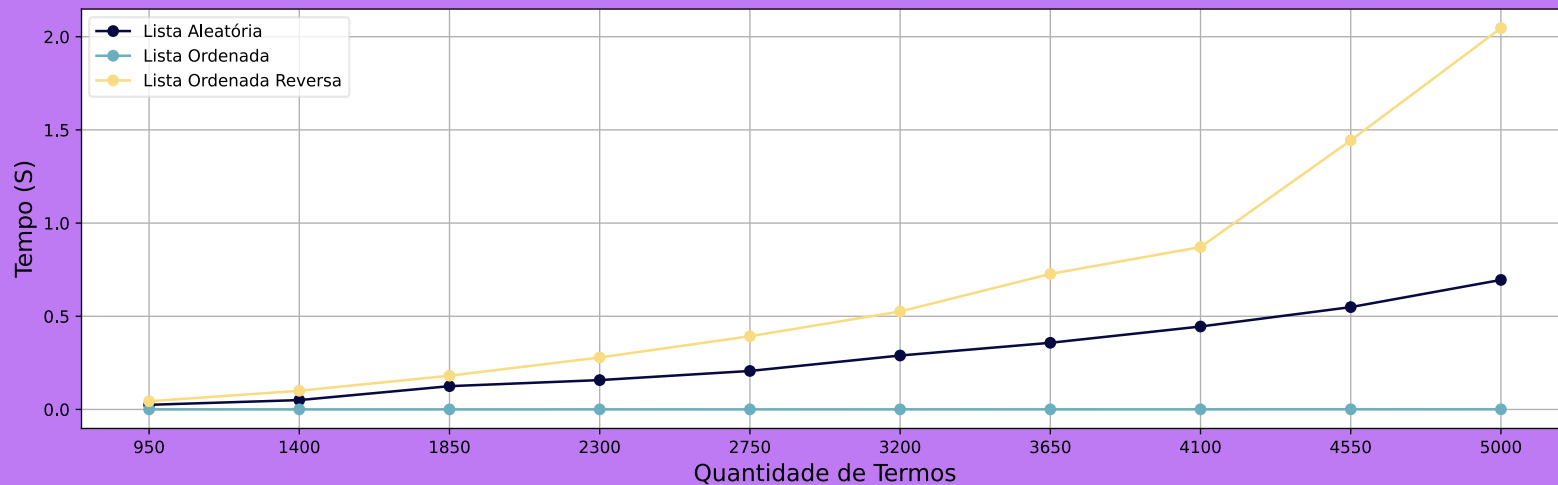
QUICK SORT

Tamanho	Tipo de Lista	Tempo (s)
950	Media Ordenada Reversa	0,001001596
1400	Media Ordenada Reversa	0,002969742
1850	Media Ordenada Reversa	0,002586365
2300	Media Ordenada Reversa	0,003000498
2750	Media Ordenada Reversa	0,005052805
3200	Media Ordenada Reversa	0,005232811
3650	Media Ordenada Reversa	0,005557537
4100	Media Ordenada Reversa	0,006030321
4550	Media Ordenada Reversa	0,008303404
5000	Media Ordenada Reversa	0,008133411

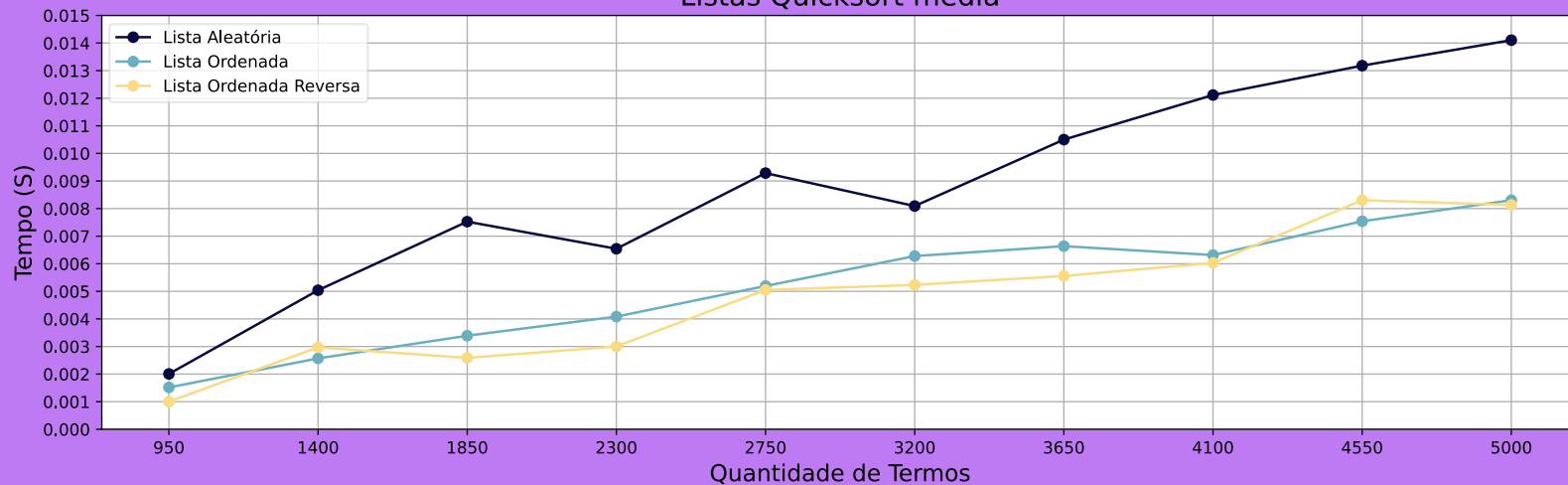
Lista Média Ordenada Reversa



Listas Média Insertion



Listas Quicksort média





RESULTADOS(ALGORITMO MAIS RÁPIDO)



INSTÂNCIA	INSERTION	QUICK
Média Aleatória		X
Média Ordenada	X	
Média Ordenada Reversa		X





INTRODUÇÃO

APLICAÇÕES

INSERTION

QUICK

DESCRIÇÕES

ANÁLISE

CONCLUSÃO

ESTRUTURA DE DADOS

3

LISTAS GRANDES



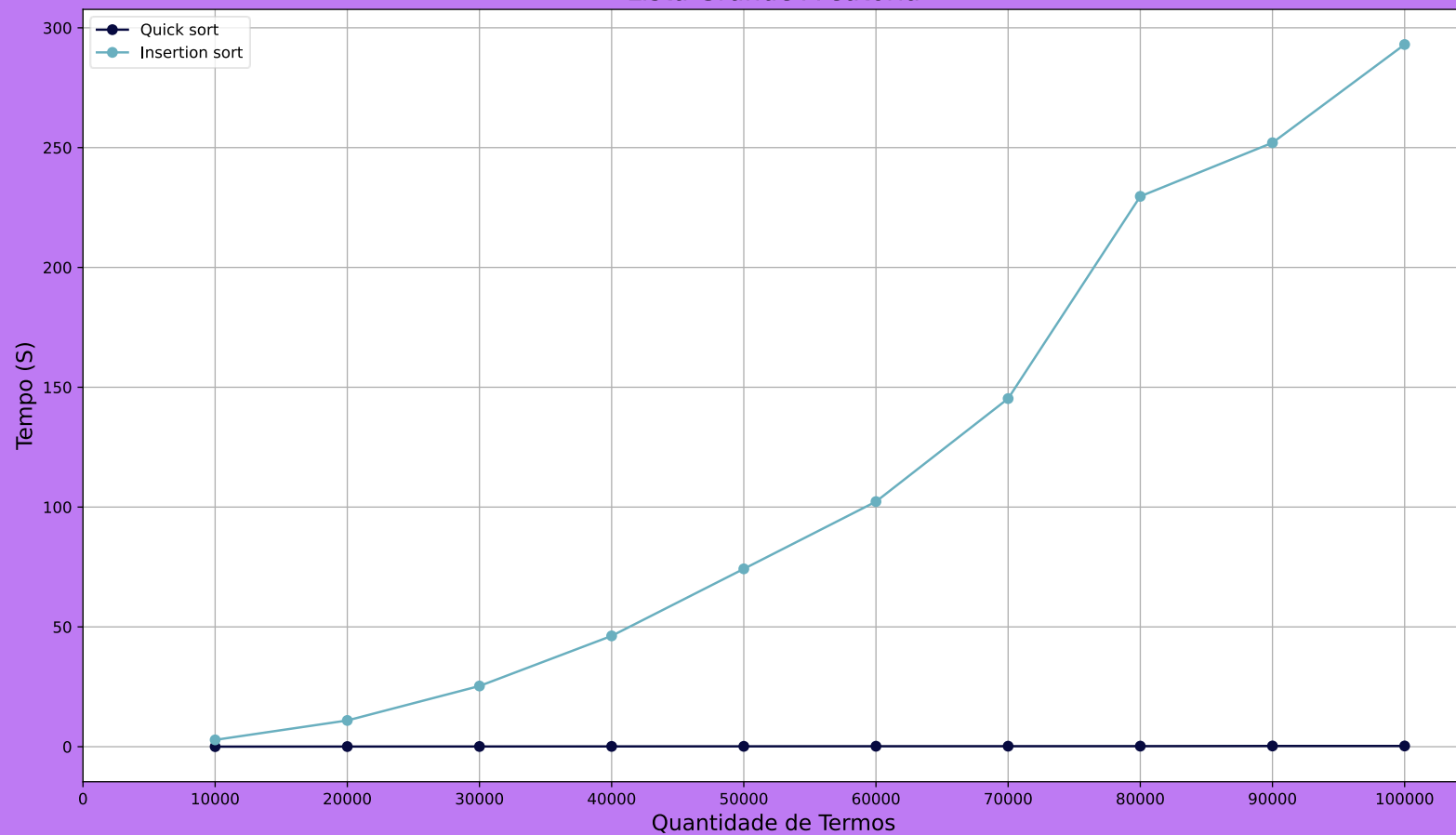
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Aleatoria	2,848602533
20000	Grande Aleatoria	10,95338559
30000	Grande Aleatoria	25,33281422
40000	Grande Aleatoria	46,24214315
50000	Grande Aleatoria	74,21604061
60000	Grande Aleatoria	102,289072
70000	Grande Aleatoria	145,2765994
80000	Grande Aleatoria	229,6628473
90000	Grande Aleatoria	252,033324
100000	Grande Aleatoria	293,0302746

QUICK SORT

Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Aleatoria	0,028579
20000	Grande Aleatoria	0,059062
30000	Grande Aleatoria	0,09041
40000	Grande Aleatoria	0,125157
50000	Grande Aleatoria	0,155709
60000	Grande Aleatoria	0,197054
70000	Grande Aleatoria	0,214945
80000	Grande Aleatoria	0,237676
90000	Grande Aleatoria	0,300369
100000	Grande Aleatoria	0,303044

Lista Grande Aleatória



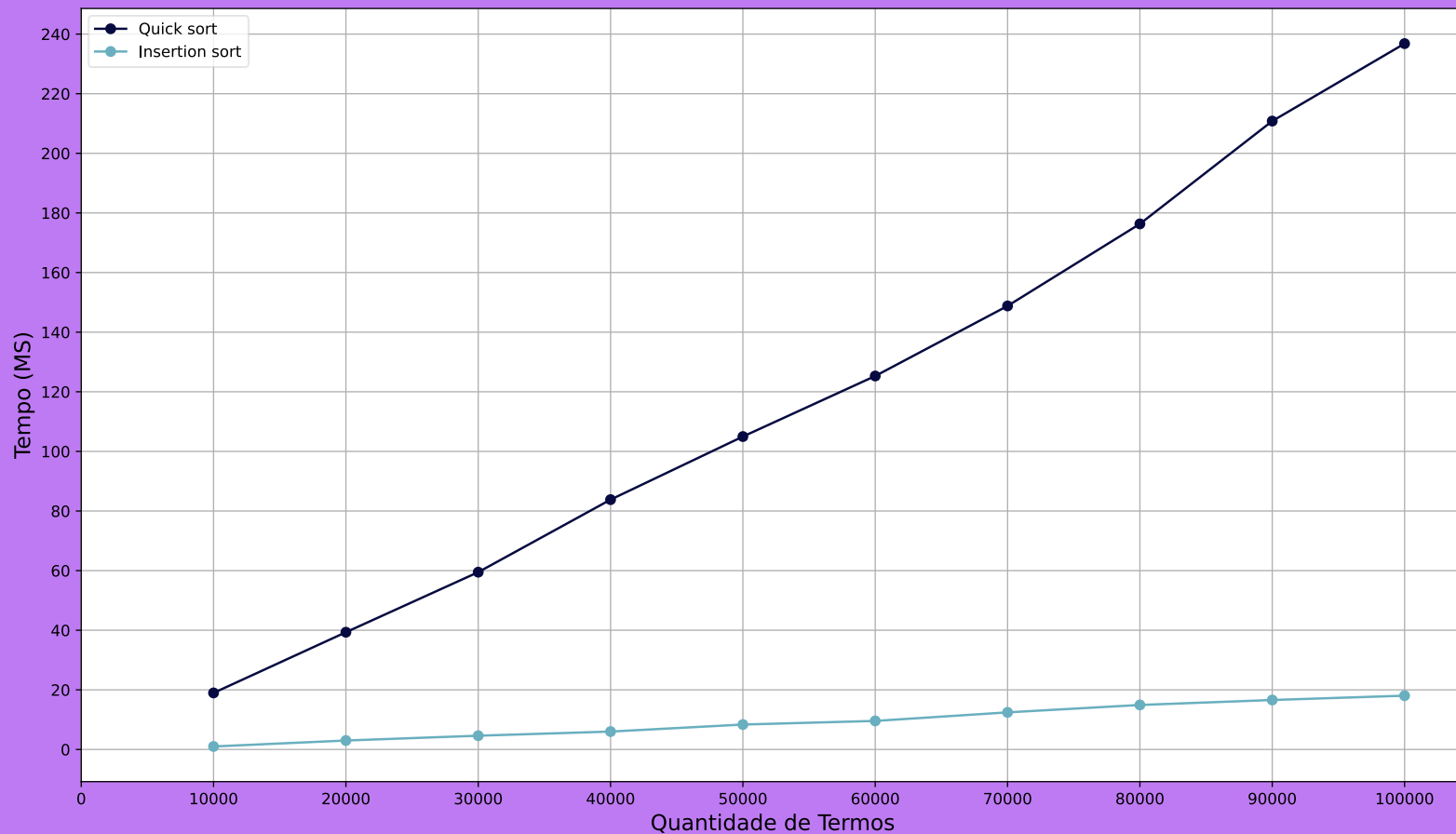
INSERTION SORT

Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Ordenada	0,001000881
20000	Grande Ordenada	0,003000736
30000	Grande Ordenada	0,004614592
40000	Grande Ordenada	0,006001234
50000	Grande Ordenada	0,008358955
60000	Grande Ordenada	0,00957489
70000	Grande Ordenada	0,012444735
80000	Grande Ordenada	0,014931679
90000	Grande Ordenada	0,016591549
100000	Grande Ordenada	0,018044233

QUICK SORT

Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Ordenada	0,018967
20000	Grande Ordenada	0,039331
30000	Grande Ordenada	0,059481
40000	Grande Ordenada	0,083812
50000	Grande Ordenada	0,104977
60000	Grande Ordenada	0,125292
70000	Grande Ordenada	0,148798
80000	Grande Ordenada	0,176327
90000	Grande Ordenada	0,210815
100000	Grande Ordenada	0,236801

Lista Grande Ordenada





INSERTION SORT

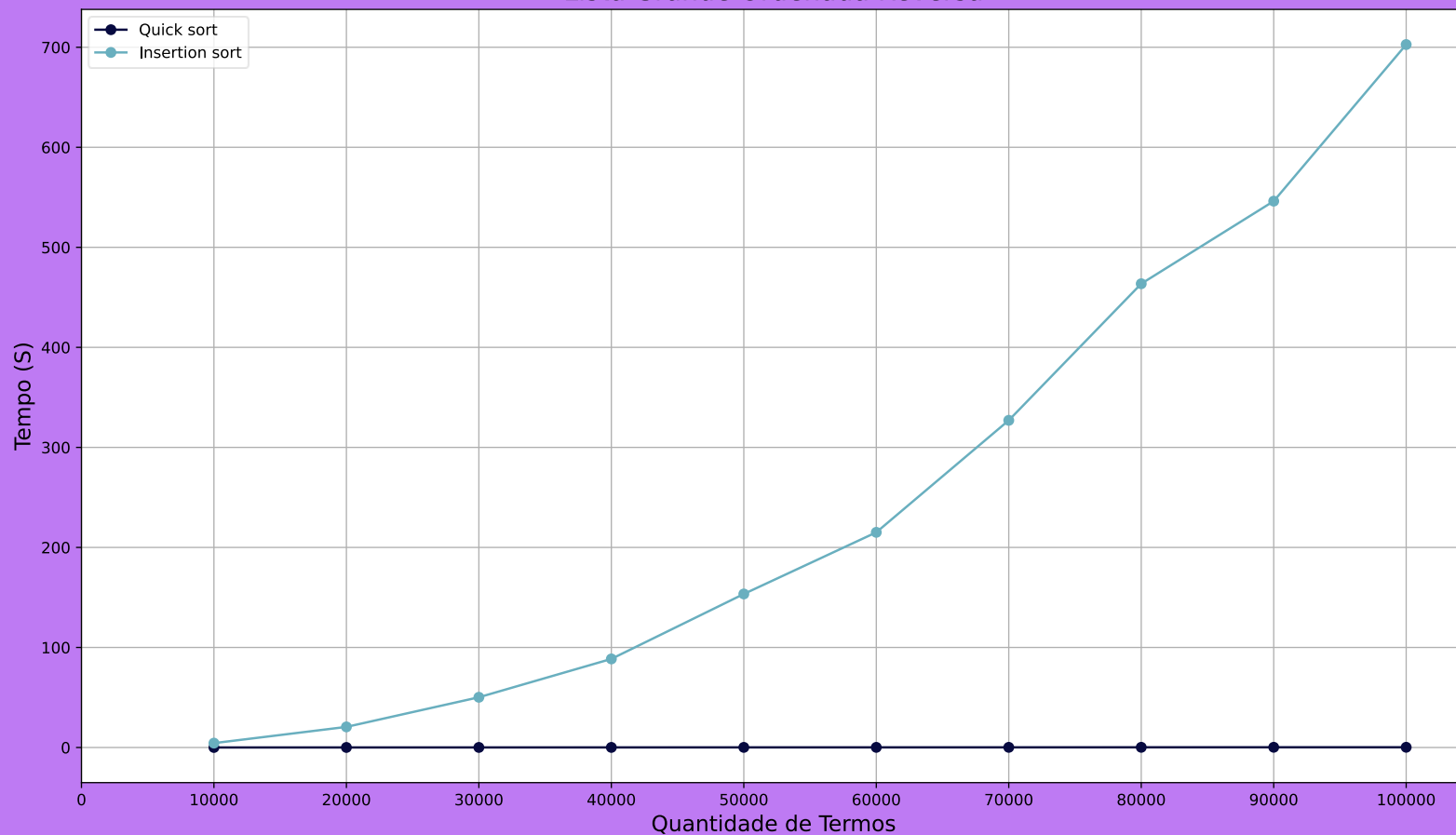
Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Ordenada Reversa	4,355406523
20000	Grande Ordenada Reversa	20,58070421
30000	Grande Ordenada Reversa	50,19435215
40000	Grande Ordenada Reversa	88,53890157
50000	Grande Ordenada Reversa	153,4700468
60000	Grande Ordenada Reversa	215,0825434
70000	Grande Ordenada Reversa	327,0374043
80000	Grande Ordenada Reversa	463,5958519
90000	Grande Ordenada Reversa	546,1598978
100000	Grande Ordenada Reversa	702,7001204



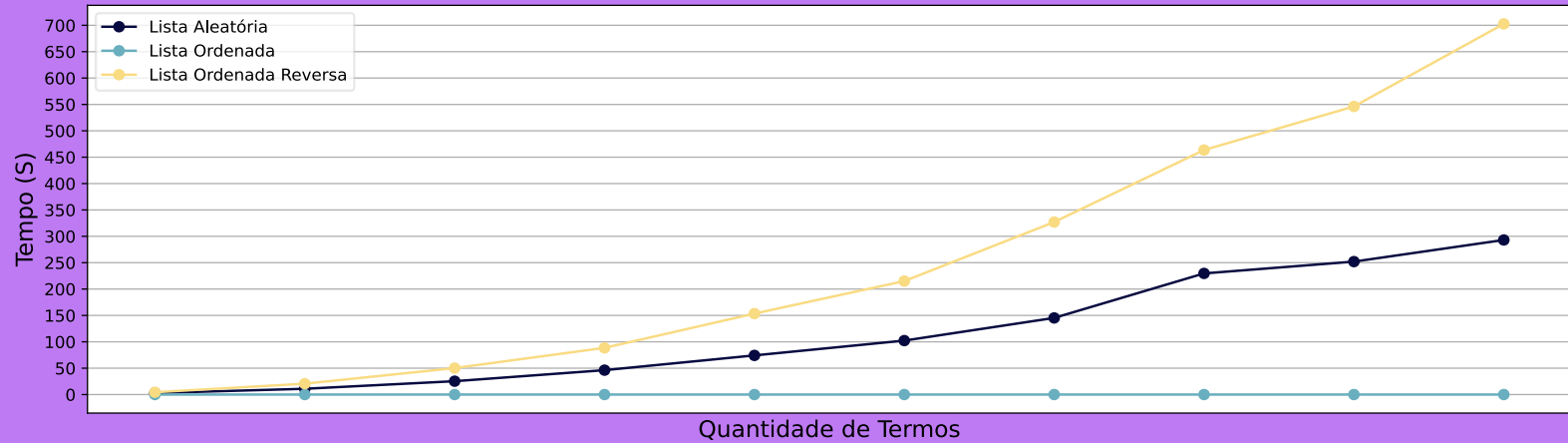
QUICK SORT

Tamanho	Tipo de Lista	Tempo (s)
10000	Grande Ordenada Reversa	0,019366
20000	Grande Ordenada Reversa	0,040923
30000	Grande Ordenada Reversa	0,063845
40000	Grande Ordenada Reversa	0,08737
50000	Grande Ordenada Reversa	0,113633
60000	Grande Ordenada Reversa	0,129078
70000	Grande Ordenada Reversa	0,14902
80000	Grande Ordenada Reversa	0,188976
90000	Grande Ordenada Reversa	0,216027
100000	Grande Ordenada Reversa	0,217232

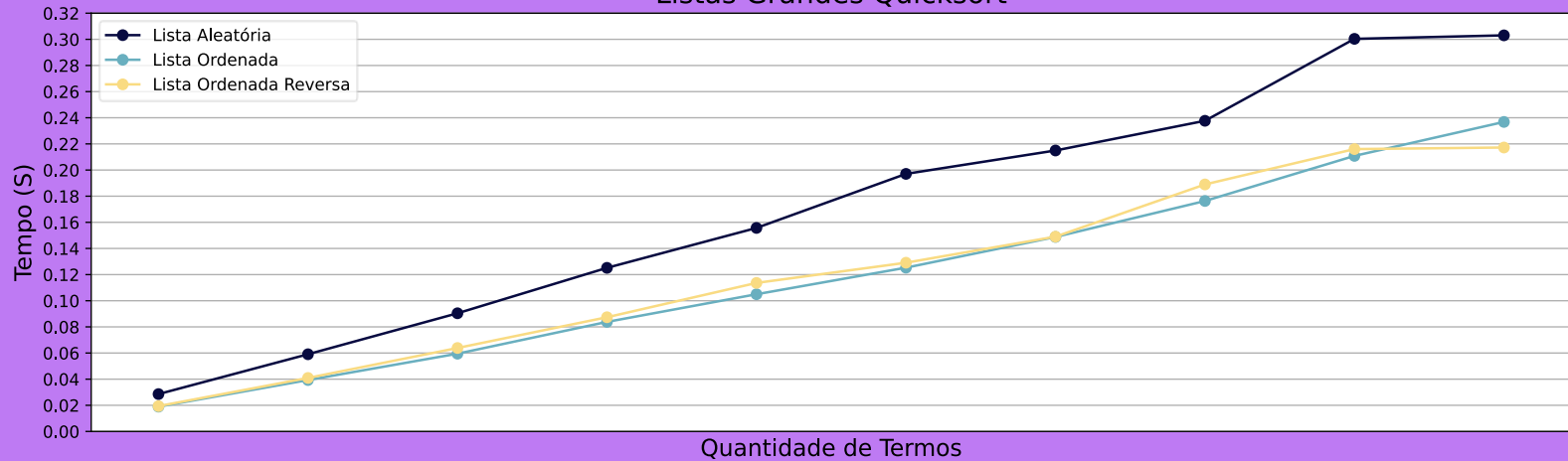
Lista Grande Ordenada Reversa



Listas Grandes Insertion



Listas Grandes Quicksort





RESULTADOS(ALGORITMO MAIS RÁPIDO)



INSTÂNCIA	INSERTION	QUICK
Grande Aleatória		X
Grande Ordenada	X	
Grande Ordenada Reversa		X





CONCLUSÃO

● LISTAS PEQUENAS

O insertion sort é mais rápido na maioria dos casos, menos quando precisa ordenar um algoritmo ordenado reversamente.

● LISTAS MÉDIA E LISTAS GRANDES

O quick sort é mais rápido na maioria dos casos, menos quando precisa ordenar um algoritmo que já esta ordenado.



[INTRODUÇÃO](#)[APLICAÇÕES](#)[INSERTION](#)[QUICK](#)[DESCRIÇÕES](#)[ANÁLISE](#)[CONCLUSÃO](#)[ESTRUTURA DE DADOS](#)

RECOMENDAÇÕES



INSERTION

Listas Pequenas.
Listas quase ordenadas.

VS

QUICKSORT

Listas médias aleatórias.
Listas grandes aleatórias.
Listas ordenadas inversamente





INTRODUÇÃO

APLICAÇÕES

INSERTION

QUICK

DESCRIÇÕES

ANÁLISE

CONCLUSÃO

ESTRUTURA DE DADOS

OBRIGADO!

