Lab_03

//----------

Prazo de entrega: 07/04/2025

Valor: 0

Não entrega: -5 valores

//----------

1. Considere a implementação do algoritmo Merge Sort, dada em aula, para ordenar um array de inteiros:

   void mergeSort(int *arr, int inicio, int fim);
   void merge(int *arr, int inicio, int meio, int fim);

   Modifique-a para contar quantas trocas e quantas comparações são feitas:

   void mergeSortContagem(int *arr, int inicio, int fim, int *troca, int *comp);
   void mergeContagem(int *arr, int inicio, int meio, int fim, int *troca, int *comp);

   Considere que:

   uma comparação ocorre sempre que dois elementos são comparados para unir os subarrays;

   uma troca ocorre sempre que as cópias dos elementos dos arrays temporários voltam para o array original.

   Simule o algoritmo com apenas 2 arrays: um com 4 e outro com 10^3 elementos.

   Considere dois arrays de entrada (os mesmos utilizados no Lab_02):

   - ordenado
   - em ordem reversa

   e armazene o número de trocas e de comparações efetuadas. Utilize o vosso array de 4 elementos para exemplificar e justificar os números de trocas e de comparações.

2. Implemente o algoritmo Quick Sort, para ordenar um array de inteiros:

   void quickSort(int *arr, int inicio, int fim, int *troca, int *comp);

   int partition (int *arr, int inicio, int fim, int *troca, int *comp);

   Considere que:

   uma comparação ocorre sempre que um elemento é comparado com o pivot;

   uma troca ocorre sempre que há rearranjo dos elementos durante a partição.

   Simule o algoritmo com apenas 2 arrays: um com 4 e outro com 10^3 elementos.

   Considere dois arrays de entrada (os mesmos utilizados no Lab_02):

   - ordenado
   - em ordem reversa

   e armazene o número de trocas e de comparações efetuadas. Utilize o vosso array de 4 elementos para exemplificar e justificar os números de trocas e de comparações.

3. O desempenho do Quick Sort depende da escolha do pivot. Investigue e discuta esta afirmação, citando o que ocorre em termos do número de comparações e de trocas. Qual seria uma boa estratégia para a escolha do pivot? Com base na vossa implementação do quick sort, apresente as vossas conclusões.

4. A linguagem C tem uma implementação do Quick Sort, na função qsort, da biblioteca stdlib.h. Use esta implementaçãp para ordenar os seus arrays e compare os tempos de execução (utilize sempre o mesmo array para efeitos de comparação entre o qsort e a vossa implementação). Se achar necessário, considere um array de tamanho maior para esta comparação.

5. Considerando as entradas (array ordenado e ordem reversa) com 10^3 elementos, qual destes algoritmos obteve a melhor performance em termos de tempo de execução? E em termos de trocas e de comparações (entre o merge sort e a vossa implementação do quick sort)? Justifique as suas respostas.

O grupo deverá escrever um relatório de, no máximo, 5 páginas sobre este experimento. O relatório deverá conter: introdução (breve explicação de cada algoritmo e as suas

complexidades no pior caso), objetivo, resultados, discussão dos resultados e conclusão.

A seção dos resultados deverá ter, obrigatoriamente, a seguinte tabela:

	Array com 10^3 elementos (n=10^3)					
	Array ordenado			Array em ordem reversa		
Algoritmo	Número de trocas	Número de comparações	Tempo de execução	Número de trocas	Número de comparações	Tempo de execução
Merge Sort						
Quick Sort (implem do grupo)						
Quick Sort (implem padrão C)	--	--		--	--	

Se julgar necessário, acrescente mais colunas para outras simulações.