

Disciplina: Estruturas de Dados I	
Aluno(a):	Nota:
Prof. Tiago Pessoa Ferreira de Lima	
Cada questão vale 2,5 pontos	
1. Implemente o algoritmo Busca Sequencial. Sua implementação deve do elemento encontrado (ou indicar ausência, se não existir) e o número realizadas durante a execução. Em seguida, analise seu desempenho, dis caso e o pior caso.	de comparações



- 2. O Bubble Sort é um dos algoritmos de ordenação mais simples. Apesar de pouco eficiente em grandes conjuntos de dados, ele é ideal para fins didáticos, pois trabalha com repetidas comparações e trocas até que a lista esteja organizada. Ele funciona assim:
 - Percorre a lista comparando pares adjacentes.
 - Se estiverem fora de ordem, troca-os.
 - Repete o processo até que nenhuma troca seja necessária.

Dada a lista inicial: [5, 3, 8, 4, 2], Complete a tabela abaixo, mostrando o estado da lista após cada passagem (não após cada troca individual).

Passagem	Estado da Lista Após a Passagem
Inicial	[5, 3, 8, 4, 2]
1	
2	
3	
4	

Escreva um pequeno relatório respondendo:

- Qual foi o número de passos necessários?
- O que aconteceria se a lista já estivesse quase ordenada?
- O Bubble Sort é eficiente para listas grandes? Por quê?

 O Bubble Sort é eficiente para listas grandes? Por quê?

 O Bubble Sort é eficiente para listas grandes? Por quê?

 O Bubble Sort é eficiente para listas grandes? Por quê?



- 3. O Insertion Sort é um dos algoritmos de ordenação mais simples. Apesar de pouco eficiente em grandes conjuntos de dados, ele é bastante utilizado em listas pequenas ou quase ordenadas, pois realiza menos comparações nesses casos. Ele funciona assim:
 - Percorre a lista da esquerda para a direita.
 - Em cada passo, "insere" o elemento atual na posição correta da parte já ordenada à esquerda.
 - Esse processo é repetido até que todos os elementos estejam na ordem desejada.

Dada a lista inicial: [5, 3, 8, 4, 2], complete a tabela abaixo, mostrando o estado da lista após cada inserção.

Inserção (Elemento)	Estado da Lista Após a Inserção
Inicial	[5, 3, 8, 4, 2]
3	
8	
4	
2	

Escreva um pequeno relatório respondendo:

- Quantos passos foram necessários para ordenar a lista?
- O que aconteceria se a lista já estivesse quase ordenada?

• O Insertion Sort é eficiente para listas grandes? Justifique sua resposta.



- 4. O Selection Sort é um algoritmo de ordenação simples e didático. Apesar de pouco eficiente em grandes conjuntos de dados, ele é fácil de entender porque segue uma ideia direta: encontrar o menor elemento e colocá-lo na posição correta, repetindo esse processo até que a lista esteja ordenada. Ele funciona assim:
 - Percorre a lista procurando o menor elemento.
 - Troca esse elemento com o da primeira posição.
 - Repete o processo para o restante da lista (da segunda posição em diante).
 - Continua até que todos os elementos estejam organizados.

Dada a lista inicial: [5, 3, 8, 4, 2], Complete a tabela abaixo, mostrando o estado da lista após cada seleção e troca.

Seleção (Posição Inicial)	Estado da Lista Após a Troca
Inicial	[5, 3, 8, 4, 2]
1^a (menor = 2)	
2^{a} (menor = 3)	
3^{a}	
4 ^a	

Escreva um pequeno relatório respondendo:

- Quantos passos foram necessários para ordenar a lista?
- O que aconteceria se a lista já estivesse quase ordenada?

•	O Selection Sort é eficiente para listas grandes? Justifique sua resposta.



Rascunho