

| Disciplina: Estruturas de Dados II |
|--|
| Aluno(a):Nota: |
| Prof. Tiago Pessoa Ferreira de Lima |
| 1. Quicksort e a ordem dos dados de entrada O desempenho do Quicksort pode variar com a ordem dos elementos na lista de entrada Explique por que isso acontece, indicando qual o pior caso para esse algoritmo e como escolha do pivô pode influenciar no tempo de execução. |
| |
| |
| |
| |
| |
| |
| 2. Merge Sort é estável? Defina estabilidade em algoritmos de ordenação. Analise se o Merge Sort é estável ilustre sua resposta com um exemplo prático em que a estabilidade faz diferença n resultado final. |
| |
| |
| |
| |
| |
| |
| |



$IFPE-{\it Campus Palmares}$

3. Mesclar listas ordenadas (sem intercalar nós)

Dada duas listas ligadas ordenadas (em ordem crescente), implemente o método mesclar_ordenadas(outra_lista) que retorna uma nova lista ligada também ordenada, contendo todos os elementos das duas listas. Não modifique as listas originais.

| Exemplo: |
|--|
| Lista A: $1 \rightarrow 4 \rightarrow 6$ |
| Lista B: $2 \rightarrow 3 \rightarrow 5$ |
| Saída esperada: nova lista $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |



4. Substituir cada elemento pela média móvel

Implemente o método substituir_por_media_movel(janela) que substitui cada elemento da lista (exceto as bordas necessárias) pela média dos valores dentro de uma janela deslizante de tamanho ímpar. Exemplo: com janela=3, cada elemento vira a média de si mesmo e seus dois vizinhos imediatos.

Exemplo com janela = 3:

Lista original: [2, 4, 6, 8, 10]

Cálculos:

- posição 1: média de (2, 4, 6) → 4.0
 posição 2: média de (4, 6, 8) → 6.0
- posição 3: média de (6, 8, 10) → 8.0

| Lista modificada: [2, 4.0, 6.0, 8.0, 10] (bordas mantidas, internos substituídos). | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| · | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |



Rascunho



| Disciplina: Estruturas de Dados II | |
|---|------------------------------|
| Aluno(a): | Nota: |
| Prof. Tiago Pessoa Ferreira de Lima | |
| 1. O Merge Sort é afetado pela ordem inicial dos elem A ordem inicial dos elementos da lista de entrada influ algoritmo Merge Sort? Justifique sua resposta com base na desse algoritmo. | encia o tempo de execução do |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |



| 2. Quicks | sor | t e um aig | gorii | tmo ae oraen | açac | estavei? | | | | | | |
|-----------|-----|------------|-------|--------------|------|------------|----|-----------|---|--------|-----|---|
| Explique | o | conceito | de | estabilidade | em | algoritmos | de | ordenação | e | avalie | se | a |
| . 1 | 4 | ~ 1 ~ | 1 | 0 1 1 / | | 1 DA | | 1 1 | | 4 | . 1 | |

| implementação padrão do Quicksort é estável. Dê um exemplo com elementos repetido para ilustrar sua resposta. | | | | | |
|---|---|--|--|--|--|
| | | | | | |
| | | | | | |
| | _ | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | _ | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | _ | | | | |
| | | | | | |
| | | | | | |



3. Intercalação de listas ligadas

Implemente o método intercalar(outra_lista) para uma lista ligada, que intercala seus nós com os da outra lista. O método deve reorganizar os ponteiros, sem criar novos nós.

Antes da intercalação:

- Lista A: $1 \rightarrow 3 \rightarrow 5$
- Lista B: $2 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8$

Depois de A.intercalar(B) (sem criar novos nós):

• Lista A modificada: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$

| | |
|---|------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| - | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |



4. Substituir por soma dos vizinhos

Implemente o método substituir_por_soma_vizinhos() em uma lista que substitui cada elemento (exceto o primeiro e o último) pela soma dos seus vizinhos imediatos.

Antes da substituição:

Lista: $2 \rightarrow 5 \rightarrow 7 \rightarrow 4 \rightarrow 9$

Durante o processamento:

- O primeiro (2) e o último (9) permanecem iguais.
- O 5 vira 2 + 7 = 9
- O 7 vira 5 + 4 = 9
- O 4 vira 7 + 9 = 16

| |
|------|
| |
| |
| |
| |
| |
| |
| |



5. Eliminação a cada N posições (variação do problema de Josephus)

Implemente o método eliminar_n_em_n(n) em uma lista circular que elimina um nó a cada n saltos, até restar apenas um. O método deve retornar o valor do último nó restante.

| Exemplo | com | valores | simp | les: |
|----------|-------|----------|---------|------|
| LACINPIC | COIII | vaioi cs | SIIII D | ics. |

- Lista inicial: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ (circular)
- Parâmetro n = 3

•

Processo de eliminação:

1. Começa no 1. Conta 1 (1), 2 (2), $3 \rightarrow$ elimina 3

$$\circ$$
 Lista: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$

2. Continua no 4. Conta 1 (4), 2 (5), $3 \rightarrow$ elimina 1

o Lista:
$$2 \rightarrow 4 \rightarrow 5$$

3. Continua no 2. Conta 1 (2), 2 (4), $3 \rightarrow$ elimina 5

$$\circ$$
 Lista: $2 \rightarrow 4$

4. Continua no 2. Conta 1 (2), 2 (4), $3 \rightarrow$ elimina 2

| | ~ | ٠ | | | |
|---------|-----|-----|-----|---|---|
| \circ | - 1 | .19 | sta | ٠ | 4 |

| - | |
|---|--|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| - | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| - | |
| | |
| | |



Rascunho