



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP3 - Primeiro Semestre de 2017

Nome -

Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (2,0) Dado um vetor V com n elementos (pode haver elementos repetidos), escreva um algoritmo que determine o elemento de V que ocorre o maior número de vezes. Havendo mais de um elemento com esta propriedade, o algoritmo deve determinar todos eles. Exemplo: se V é formado pelos elementos 7, 3, 6, 7, 1, 6, 5, 9, 6, 8, 8, 7, a resposta deve ser: 6 e 7 (pois ocorrem 3 vezes cada). Qual é a complexidade do seu algoritmo?

Resposta: O Algoritmo 1 retorna uma fila auxiliar F contendo todos os elementos mais frequentes de V . O algoritmo inicialmente ordena V , fazendo com que os elementos repetidos se agrupem de forma crescente. A fila F é criada de modo a armazenar os elementos mais frequentes de V a cada iteração do algoritmo. A variável atual guarda o valor do elemento comparado na iteração anterior com a corrente, de modo que seu valor é alterado somente quando um novo elemento é encontrado no vetor. A variável freq_atual guarda o número de repetições da variável atual, enquanto maior_freq guarda o valor máximo dentre as frequências encontradas até então. Sempre que um elemento i possui frequência maior que a máxima, F é esvaziada, fazendo com que seus ponteiros de entrada e saída sejam os mesmos, e é feita a inserção de i em F . Caso a frequência do elemento corrente i seja igual à máxima, somente é feita a inserção de i em F . Pode-se ver que a complexidade do algoritmo é dominada pela operação de ordenação de V , cuja ordem é $O(n \log n)$, uma vez que as operações de inserção e esvaziamento da fila são feitas em tempo constante.

O algoritmo anterior possui a mesma complexidade para qualquer vetor de entrada V . Porém, caso se saiba que a diferença entre os valores mínimo e máximo de V é da ordem de n , então pode-se usar uma tabela de dispersão por encadeamento exterior para os elementos de V e uma contagem simples do tamanho de cada lista da tabela. Com isso obtemos um algoritmo linear no tamanho de V . Note que o mesmo algoritmo se aplicado ao caso geral possui complexidade $O(n \cdot M)$, onde M é o valor do maior elemento de V , uma vez que a tabela terá M posições.

2. (2,0) Considere uma lista simplesmente encadeada ordenada contendo os nós com os valores: 12, 23, 34, 45, 56, 67. Desenhe esta lista, representando todos os ponteiros. Redesenhe a lista após a remoção do nó 34, mostrando as alterações feitas nos ponteiros.

Resposta: A Figura 1 representa tais operações.

3. (2,0) Duas árvores binárias de prefixo são ditas similares quando uma delas pode ser obtida da outra realizando trocas de subárvores. (Uma troca de subárvore consiste em escolher um nó x e trocar sua subárvore esquerda com a direita.) Desenhe duas árvores de Huffman não similares relativas às frequências 2, 2, 4, 4.

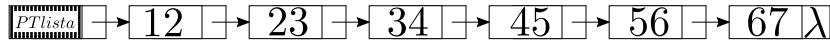
Resposta: A Figura 2 representa tais árvores.

Algoritmo 1: *Mais_Frequentes(V)*.

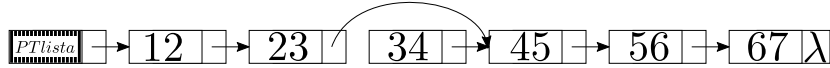
Entrada: Vetor V não ordenado composto por $n \geq 1$ valores.

Saída: Os valores dos elementos mais frequentes de V .

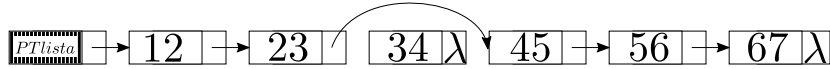
```
1  $V \leftarrow \text{Ordenação}(V)$ ;  
2  $F \leftarrow \text{NovaFila}()$ ;  
3  $\text{atual} \leftarrow V[1]$ ;  
4  $\text{freq\_atual} \leftarrow 1$ ;  
5  $\text{maior\_freq} \leftarrow \text{freq\_atual}$ ;  
6  $\text{Inserir}(V[1], F)$ ;  
7 para  $i \leftarrow 2, \dots, n$  faça  
8   se  $V[i] = \text{atual}$  então  
9      $\text{freq\_atual} \leftarrow \text{freq\_atual} + 1$ ;  
10  senão  
11    se  $\text{freq\_atual} > \text{maior\_freq}$  então  
12       $\text{maior\_freq} \leftarrow \text{freq\_atual}$ ;  
13       $\text{Esvaziar}(F)$ ;  
14       $\text{Inserir}(V[i], F)$ ;  
15    senão  
16      se  $\text{freq\_atual} = \text{maior\_freq}$  então  
17         $\text{Inserir}(V[i], F)$ ;  
18     $\text{atual} \leftarrow V[i]$ ;  
19     $\text{freq\_atual} \leftarrow 1$ ;  
20 retorna  $F$ ;
```



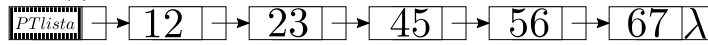
(a) Representação da lista simplesmente encadeada e ordenada.



(b) Mudança do ponteiro próximo do nó com elemento 23 para o que contém 45.



(c) Mudança do ponteiro próximo do nó contendo o elemento 34 para nulo.



(d) Eliminação do nó contendo o elemento 34.

Figura 1: Sequência de operações realizadas para eliminar um nó de uma lista simplesmente encadeada.

4. (2,0) Considere o Algoritmo de Força Bruta para Casamento de Cadeias. Quantas comparações entre caracteres este algoritmo faz ao procurar a

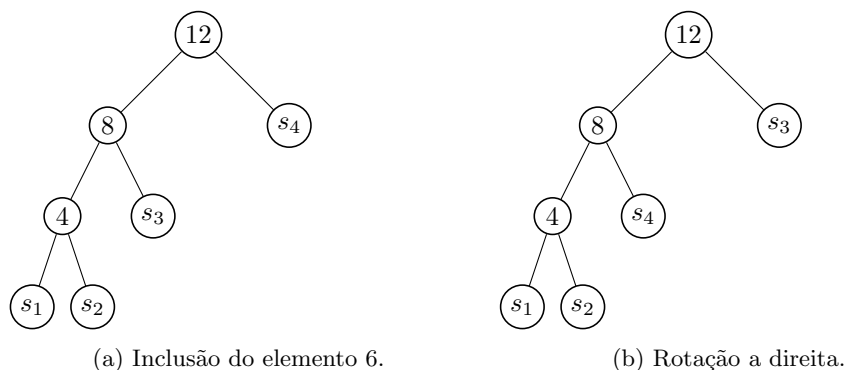


Figura 2: Árvores de Huffman não similares.

cadeia **aaabb** dentro da cadeia **aaaaaaabb** ? Responda: este exemplo corresponde ao pior caso do algoritmo? Justifique.

Resposta: O algoritmo de força bruta compara o casamento da subsequência de menor tamanho, digamos m , para cada posição da sequência de tamanho maior, digamos n , até a posição $n - m$. Dessa forma, no exemplo dado são feitas quatro comparações em cada sub-cadeia iniciando com três a 's, exceto para os últimos três a 's, onde a cadeia menor é encontrada como um sufixo da maior, que possui comprimento 5. Assim efetua-se $(4 \times 4) + 5 = 21$ comparações no total.

Este exemplo não corresponde ao pior caso, uma vez que não é feita a comparação completa de cada sub-cadeia menor. O pior caso ocorre quando a comparação falha apenas no último caractere da cadeia menor, para cada posição da cadeia maior.

5. (2,0) Aplique o método de ordenação por bolhas (Bubblesort) ao vetor abaixo, de modo que ele fique ordenado decrescentemente, isto é, o maior valor fica à esquerda e o menor valor à direita. Mostre todas as trocas de posição entre elementos.

32 33 27 31 29 26 25 30 28

Resposta: Abaixo seguem as trocas.

–Trocas pelo Método da Bolha: são efetuadas 10 trocas.

32* 33 27 31 29 26 25 30 28 Vetor inicial.

32 33* 27 31 29 26 25 30 28

33* 32 27 31 29 26 25 30 28

33 32 27* 31 29 26 25 30 28

33 32 27 31* 29 26 25 30 28

33 32 31* 27 29 26 25 30 28

33 32 31 27 29* 26 25 30 28

33	32	31	29*	27	26	25	30	28
33	32	31	29	27	26*	25	30	28
33	32	31	29	27	26	25*	30	28
33	32	31	29	27	26	25	30*	28
33	32	31	29	27	26	30*	25	28
33	32	31	29	27	30*	26	25	28
33	32	31	29	30*	27	26	25	28
33	32	31	30*	29	27	26	25	28
33	32	31	30	29	27	26	25	28*
33	32	31	30	29	27	26	28*	25
33	32	31	30	29	27	28*	26	25
33	32	31	30	29	28*	27	26	25
33	32	31	30	29	28	27	26	25 Vetor ordenado.