

Estruturas de Dados e Algoritmos II

1ª Frequência

Departamento de Informática
Universidade de Évora

7 de Abril de 2017

1. [3,5 valores] Assumindo que o alfabeto consiste nas 26 letras minúsculas, desenhe uma *trie* cujo conteúdo sejam as cinco palavras

basta doze bar barco mar

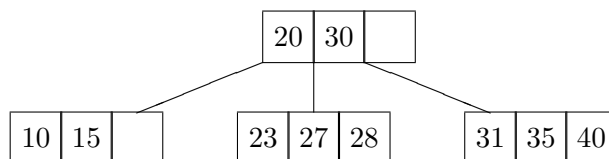
Qual seria a memória ocupada por uma implementação em C da *trie* que desenhou, numa máquina com endereços e palavras de 32 bits? (Não precisa de calcular o valor, mas apresente e justifique todos os cálculos efectuados ou a efectuar.)

2. [2 valores] Qual o número máximo de nós de uma *trie* com três elementos, de comprimentos 3, 4 e 5? E o número mínimo? Justifique as suas respostas.

3. [3,5 valores] A *B-tree* da figura tem grau de ramificação mínimo 2. Apresente o seu estado depois da execução de *cada uma* das operações da sequência

i 12 **r** 28 **r** 30 **i** 19 **i** 11 **i** 38 **r** 10

pela ordem apresentada. As letras **i** e **r** indicam, respectivamente, a inserção e a remoção do elemento que se lhes segue.



4. [2,5 valores] Apresente o pseudo-código da função B-TREE-SIZE(x), que devolve o número de nós da *B-tree* cuja raiz é o nó x (que já está em memória).

Considere que os nós de uma *B-tree* têm os campos **n** (ocupação), **c** (filhos), **key** (elementos) e **leaf** (é folha?).

5. [2 valores] Calcule o número máximo de elementos de uma *B-tree* de grau de ramificação mínimo 6 cuja altura é 3. (Não precisa de calcular o valor, mas apresente e justifique todos os cálculos efectuados ou a efectuar.)

(CONTINUA...)

6. [3 valores] O JOGO A FEIJÕES é jogado por dois jogadores que retiram, alternadamente, um montinho de feijões de uma das extremidades de uma sequência de montinhos de feijões. O jogo acaba quando é retirado o último montinho de feijões e o vencedor é o jogador que conseguiu acumular mais feijões no decorrer do jogo.

A tabela abaixo mostra várias possibilidades (não todas) de jogadas que poderiam ser feitas pelos jogadores A e B durante um jogo em que a sequência inicial consistisse em montinhos com 5, 8, 3 e 1 feijões, por esta ordem.

Feijões retirados				Total de feijões		Diferença	Vencedor
A	B	A	B	A	B	A – B	
5	8	3	1	8	9	–1	B
5	1	8	3	13	4	9	A
1	5	8	3	9	8	1	A
1	3	8	5	9	8	1	A

O primeiro jogador, o A, pode, na sua primeira jogada, retirar o monte com 5 ou o monte com 1 feijões. Se retirar o com 5, o jogador B, na sua vez, pode optar por retirar o com 8 ou o com 1 feijões. Se optar pelo primeiro, A poderá retirar o monte com 3 feijões, deixando o monte com 1 único feijão para a última jogada de B. Se for esta a sequência de jogadas, a diferença entre o número de feijões retirados por A e o número de feijões retirados por B será $(5 + 3) - (8 + 1) = -1$, o que significa que B vence o jogo.

Se A, na sua primeira jogada, optar por retirar o monte com 1 feijão, então, quer B escolha o monte com 3 ou o monte com 5 feijões, A poderá retirar o monte com 8 na jogada seguinte e vencer o jogo, já que a diferença entre o número de feijões retirados por cada jogador é de 1, a favor de A.

Apresente *uma função recursiva* que, dada uma sequência não vazia de inteiros positivos f_1, f_2, \dots, f_n , correspondentes ao número de feijões em cada montinho, calcule o melhor resultado que o primeiro jogador pode obter, em termos da diferença entre o número de feijões retirados pelo primeiro jogador e o número de feijões retirados pelo segundo, assumindo que este também tentará usar a melhor estratégia durante o jogo.

Indique claramente o que representa cada uma das variáveis que utilizar e explicita a chamada inicial. (Note que não é pedido que escreva código.)

7. [3,5 valores] Considere a função recursiva $L[i]$, onde:

- $V = v_1, v_2, \dots, v_n$ é uma sequência não vazia de inteiros não negativos;
- $P = p_1, p_2, \dots, p_n$ é uma sequência não vazia de inteiros positivos; e
- $0 \leq i \leq n$; e
- $0 \leq j \leq c$, com c um inteiro positivo.

$$L[i, j] = \begin{cases} 0 & \text{se } i = 0 \vee j = 0 \\ L[i - 1, j] & \text{se } i, j > 0 \wedge p_i > j \\ \max \left\{ v_i + L[i - 1, j - p_i], L[i - 1, j] \right\} & \text{se } i, j > 0 \wedge p_i \leq j \end{cases}$$

Apresente o pseudo-código de um algoritmo iterativo que, dados c e as duas sequências não vazias V e P , calcula o valor de $L[n, c]$. Estude (justificando) a complexidade temporal do seu algoritmo.