

Estruturas de Dados e Algoritmos II

1ª Frequência

Departamento de Informática
Universidade de Évora

5 de Abril de 2019

O enunciado deve ser entregue devidamente identificado.

Nome: _____ Número: _____

1. [4 valores] Assumindo que o alfabeto consiste nas 26 letras maiúsculas, desenhe uma *trie* cujo conteúdo sejam as seis palavras:

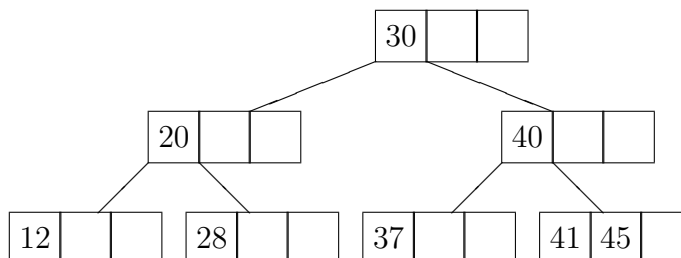
BOCA BECA TOCA E ESTA ESSES

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. [4 valores] A *B-tree* da figura tem grau de ramificação mínimo 2. Apresente o seu estado depois da execução de *cada uma* das operações seguintes, *em sequência*, pela ordem apresentada:

i 13 **r** 45 **r** 30 **r** 41 **i** 18 **i** 7 **r** 12 **i** 23

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



3. [3 valores] Apresente o pseudo-código da função B-TREE-LEAVES(x), que calcula e devolve o número de folhas 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?).

4. [2 valores] Diga, justificando a sua resposta, quantos nós de uma *B-tree* de altura *h* são acedidos, no melhor e no pior casos, durante a inserção de um elemento. (Os nós acedidos incluem os previamente existentes e os eventualmente criados durante a aplicação do algoritmo.)

(CONTINUA...)

5. Numa das provas de um concurso, os concorrentes devem fazer um percurso, saltando numa sequência de trampolins. Quando um concorrente salta num trampolim, a força de impulsão do trampolim determina sobre quantos trampolins da sequência o concorrente saltará até cair no próximo. (Se a força de impulsão do trampolim i for s_i , o trampolim onde o concorrente cairá, depois de saltar no i , será o $i + s_i + 1$.) Em cada trampolim, o concorrente pode escolher saltar ou passar ao próximo da sequência sem saltar.

A cada trampolim está associado um valor e o prémio dos concorrentes é a soma dos valores dos trampolins onde saltaram.

Sejam n (um inteiro positivo) o número de trampolins, e $V = (v_1 v_2 \dots v_n)$ e $S = (s_1 s_2 \dots s_n)$ (duas sequências de inteiros não negativos), respectivamente, os valores associados aos trampolins e a força de impulsão dos trampolins.

A função recursiva $p_{V,S}[i]$, abaixo, calcula o prémio máximo que um concorrente pode ganhar, se os trampolins onde pode saltar forem $\{i, i + 1, \dots, n\}$, com i um inteiro positivo.

$$p_{V,S}[i] = \begin{cases} 1 & \text{se } i = n + 1 \\ \max \{ v_i, p_{V,S}[i + 1] \} & \text{se } i \leq n \text{ e } i + s_i > n \\ \max \{ v_i + p_{V,S}[i + s_i + 1], p_{V,S}[i + 1] \} & \text{se } i \leq n \text{ e } i + s_i \leq n \end{cases}$$

- (a) [1 valor] Apresente a chamada da função que permite calcular o prémio máximo que um concorrente pode ganhar se puder saltar toda a sequência de trampolins.
- (b) [4 valores] Apresente o pseudo-código de um algoritmo iterativo que, dado um inteiro positivo k , calcula o valor de $p_{V,S}[k]$. Estude (justificando) as complexidades temporal e espacial do seu algoritmo.
- (c) [2 valores] Altere o pseudo-código que escreveu de modo a se poder saber em que trampolins um concorrente pode saltar para obter o prémio máximo. Escreva o pseudo-código de uma função que, dadas as sequências V e S , apresente o valor do prémio máximo e uma sequência de trampolins a saltar para obter esse prémio.