1º Exame ou 2º Teste de Estrutura de Dados e Algoritmos II

8/6/2016 - duração: 2 hora (9:00 às 11:00)

Atenção: Exame - grupo I,II,III,IV — 2º Teste - grupo III,IV e V.

Grupo I Considere o seguinte tipo para representar Tries em C:

```
struct trie {int existe; struct trie *prox[Nel];};
```

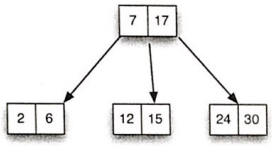
1. Desenhe a Trie sobre alfabeto {a ... z}, com as chaves: "afec", "abce", "bcf", "afece", "afecc", "abcf", "bcb"

2. Qual é o espaço ocupado pela estrutura anterior considerando que um `int` ocupa 4 bytes e que um `struct trie *` também ocupa 4 bytes.

3. Defina a função *quantosPref* que dada uma trie e uma string s retorna o número de palavras que têm a string como prefixo, (para a trie alínea a) se s fosse "af" esta função retornava: 3"

   Indique, justificando, a complexidade temporal da sua função.

Grupo II Considere a Btree de ordem t=2.



1. Indique o resultado de inserir as seguintes chaves na btree da figura acima: 25, 14, 3, 4, 13 e 26.

2. Indique o resultado de remover as seguintes chaves na btree da figura acima: 7,6, 12

3. Qual é a altura máxima de uma btree de ordem 5 com 345 chaves. Justifique.

4. Defina um tipo em C para representar a Btree acima. Indique qual o espaço ocupado pela estrutura que representa a Btree no tipo que propôs.

5. Defina uma função que dada uma Btree, retorna 1 sse todas as folhas da Btree têm a mesma profundidade.

   Indique, justificando, a complexidade temporal da sua função.

Grupo III Suponha que usa o tipo de dados abaixo para representar grafos.

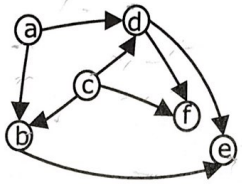
1. Indique justificando o espaço necessário (em bytes) para representar o grafo da figura.

   (Considere que um `int` ocupa 4 bytes, e que todos os outros tipos e todos os apontadores ocupam 4 bytes.)

2. Defina a função *ExisteCaminho* que dado um grafo, para todos os seus nós *u,v* sempre que existe um caminho de u para v acrescenta um arco entre u e v (se ainda não existir). (Para o grafo da figura seriam acrescentados os arcos: (a,e), (a,f) e (c,e).

   Assuma que tem a função pesquisa em largura (bfs) definida. E para representar o grafo, use os seguintes tipos:

$(6 \times 4 \times 5) + (8 \times 4 \times 3)$




```
struct lista{struct vertice  *v;
 int d;
 struct lista *prox;};
enum Cor {branco,cinzento,preto};
struct vertice {enum Cor cor;
                int distancia;
                int v;
                struct vertice *pai;
                struct lista *adj;};
void bfs(struct vertice *g, struct vertice *s, int nvertices);
```

→Faça a análise de complexidade da sua definição da função *ExisteCaminho*.

Grupo IV

Suponha que pretende encontrar o shift (deslocamento) do padrão p1 no texto t1.
p1 = 00101
t1 = 10100100101001011̲0

1. Simule os passos do algoritmo de Rabin-Karp indicando o número de comparações necessárias para encontrar a primeira coincidência. (Indique a codificação do texto e do padrão)

2. Construa o autómato finito que reconhece o padrão p1. Indique, justificando, quantas comparações são necessárias para encontrar a primeira ocorrência de p1 em t1 com o autómato finito.

3. Suponha que quer encontrar o indices das n ocorrências do padrão p1.num texto sobre o alfabeto {0,1}. Defina uma função *ncorrencias* que dada a função de transição de estados do padrão p1 e um n inteiro, devolve o indice das primeiras n ocorrências seguidas de p1. Sugestão: altere o algoritmo de cálculo do indice de ocorrência com autómatos finitos.

   Indique justificando a complexidade da sua função (número de comparações).

4. Defina a função maiorPrefixo que dadas duas strings s1 e s2, retorna o maior prefixo de s1 que é prefixo de s2.

Grupo V Considere o seguinte texto: "o potro da pata torta trota pelo porto"

1. Construa a árvore de Huffman para este texto. E codifique os símbolos do texto (note que o ' ' (espaço) também é um símbolo).

2. Defina um tipo para representar a árvore de Huffman e defina a função constroi_ah que dada uma tabela de simbolos e frequências, retorna a ávore de Huffman.

   Indique justificando a complexidade temporal da sua função.