2^o Teste+Exame de Análise e Desenho de Algoritmos

25/6/2003

duração: 2.30 horas (17:00 às 19:30)

Atenção: preencha o número e entregue o enunciado

2^o Teste:

Grupos	3	4	5
Alíneas	a, b, c	a, b, c	a, b, c

Exame:

Grupos	1	2	3	4
Alíneas	a, b	a, b	b, c	b, c

Número:________________

1. Considere o seguinte tipo para representar Tries em C:

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

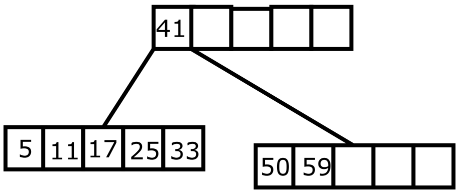
   (a) (1.5) Desenhe a Trie sobre alfabeto $\{a \ldots z\}$, com as chaves: "abc", "abcb", "abccae", "abcd", "abce"

   (b) (3.5) Defina a função *maisLonga*(struct trie *t, char *c) devolve em c a string mais longa (com mais caracteres) que está na trie t. Indique, justificando, a complexidade da sua função.

   ex: "abccae" é a string mais longa da trie desenhada na alínea anterior.

2. Considere o seguinte tipo para representar Btrees e a Btree de ordem t=3.

   ```
   struct btree {
     int folha;
     int n;
     int chave[maxchaves];
     struct btree *filho[maxchaves+1];
   };
   ```



   (a) (1.5) Indique o resultado das seguintes operações na btree da figura acima: inserir 2; remover 33. (primeiro insira 2 e depois remova o 33).

   (b) (3.5) Defina a função *lista nprimeiras* que dada uma Btree e um inteiro n, lista as n primeiras chaves da Btree. Se a Btree só tiver m chaves, com $m < n$ liste as m chaves. Analise a complexidade da sua função.

3. Suponha que pretende encontrar o shift (deslocamento) do padrão p1 no texto t1.

   p1 = 00101

   t1 = 101001001010010110

   (a) (2) Simule os passos do algoritmo de Rabin-Karp indicando o número de comparações necessárias para encontrar a primeira coincidência.

   (b) (2) Construa o autómato finito que reconhece o padrão p1. Indique, justificando, quantas comparações são necessárias para encontrar o primeiro shift de p1 em t1 com o autómato finito.

   (c) (3) Suponha que quer encontrar o shift de n ocorrências seguidas 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 shift das primeiras n ocorrências seguidas de p1. Sugestão: altere o algoritmo de cálculo de shift com autómatos finitos. Indique justificando a complexidade da sua função (número de comparações).

4. Considere o grafo da figura ao abaixo.

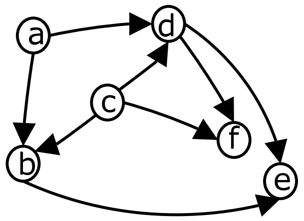
   (a) (1) Desenhe a floresta de árvores do percurso em profundidade (dfs) deste grafo.

   (b) (1.5) Desenhe a árvore do percurso em largura (bfs) com inicio no nó *a*.

   (c) (3.5) Defina a função *fechoTransitivo* 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:

```
 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);
```



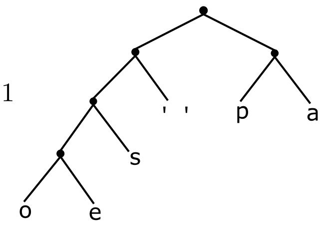
   Faça a análise de complexidade da sua definição da função *fechoTransitivo*.

5. Considere o seguinte texto: “o potro da pata torta trota pelo porto”

   (a) (2) Construa a árvore de Huffman para este texto. E codifique os símbolos do texto (note que o ' ' (espaço) também é um símbolo).

   (b) (2) Use a árvore de huffman, na figura ao lado, para descodificar o texto abaixo:

   00100010100000110111011011011101100100100010110111011



   (c) (3) Proponha um tipo em C para representar uma árvore de huffman. Defina uma função *listaSimb* que dada uma árvore de huffman lista os símbolos e os códigos.

   Indique, justificando, a complexidade da função que definiu.