Métodos de Programação II

LESI / LMCC

22 de Julho 2003

Questão 1 [*Análise de Algoritmos*]
Considere o seguinte algoritmo para o problema da exponenciação módulo um número inteiro: $y = x^b$ `%` p.

```
int modExp (int x, int p, int b[], int l)
{
  int i,y;
  y=1;
  for (i=l-1; i>=0 ; i--) {
    y=y*y % p;
    if (b[i]==1) y=y*x % p;
  }
  return y;
}
```

Considere ainda que:

- todos os números têm `l` bits.
- $b = b_{l-1}2^{l-1} + b_{l-2}2^{l-2} + \ldots + b_1 2^1 + b_0$, sendo `b[0 ... l-1]` a representação binária do expoente,
- `%` representa o resto da divisão inteira,
- as operações `*` e `%` têm um custo $O(l^2)$
- as operações `+` e `-` têm um custo $O(l)$

Responda às seguintes questões, tendo em atenção que elas podem ser respondidas por qualquer ordem:

1. Quantas operações de soma, multiplicação e divisão efectua este algoritmo no melhor e no pior caso? E quantas vezes executa a operação `if`?
2. Numa avaliação do tempo de execução de um qualquer algoritmo de exponenciação, como definiria o tamanho do input? Justifique e estabeleça um paralelo com este algoritmo em particular.
3. Caracterize o comportamento assimptótico deste algoritmo utilizando as notações O, Ω e Θ. Justifique a sua resposta e retire conclusões quanto à informação fornecida por estas notações, nomeadamente no que respeita a factores de escala e a detalhes para pequenos inputs.
4. Demonstre a correcção do algoritmo utilizando o seguinte invariante de ciclo:

   No início de cada iteração, temos $y = x^e$ `%` p, *com*

   $$e = b_{l-1}2^{l-2-i} + b_{l-2}2^{l-3-i} + \ldots + b_{i+1}$$

Nome:_____ **Número:**_____ **Curso:**_____

Métodos de Programação II

LESI / LMCC

22 de Julho 2003

Questão 2 [*Análise de Algoritmos*] Considere o seguinte algoritmo para o cálculo de números de Fibonacci:

```
int fib (int n)
{
  if (n==0 || n==1) return 1;
  else return fib(n-1) + fib(n-2);
}
```

Apesar de traduzir exactamente a definição da sequência de números de Fibonacci, este algoritmo é muito ineficiente (de tempo exponencial). Assuma que as operações aritméticas elementares se efectuam em tempo $\mathcal{O}(1)$.

1. Escreva uma recorrência que descreva o comportamento temporal do algoritmo. Desenhe a respectiva árvore de recursão para $n = 5$.
2. Prove que $T(n) = \mathcal{O}(2^n)$.
3. Prove que $T(n) = \Omega(n^2)$.
4. Escreva em **C** um algoritmo alternativo mais eficiente. Analise o seu tempo de execução.
5. Prove, para o algoritmo recursivo original, que $T(n) = \Omega(n^2)$.

Nome:_____ **Número:**_____ **Curso:**_____

Métodos de Programação II

LESI / LMCC

22 de Julho 2003

Questão 3 [*Árvores binárias e AVL*]
Considere as seguintes declarações:

```
typedef struct node {
   int data;
   struct node* lptr;
   struct node* rptr;
} Node;

typedef Node *Tree;

int ordem(Tree t, int k);
int gama(Tree t, int a, int b);
Tree merge(Tree a, Tree b);
```

1. Implemente a função `ordem` que retorna o elemento de ordem `k` existente na árvore i.e. retorna o menor elemento tal que existem $k-1$ elementos de valor inferior.

2. Explique porque é que o problema da alínea anterior não pode ser resolvido em menos do que $O(2N) = O(N)$. Forneça uma definição alternativa do tipo `Tree` que permita implementar esta função em $O(lg$ n).

3. Implemente a função `gama` que imprime os elementos da árvore com valores $a \leq x \leq b$. O tempo de execução da sua função deverá ser $O(n+N)$ em que n é o número de elementos imprimidos, e N o número total de elementos na árvore.

4. Implemente a função `merge` que funde duas árvores binárias de pesquisa numa só.

Nome:_____ **Número:**_____ **Curso:**_____

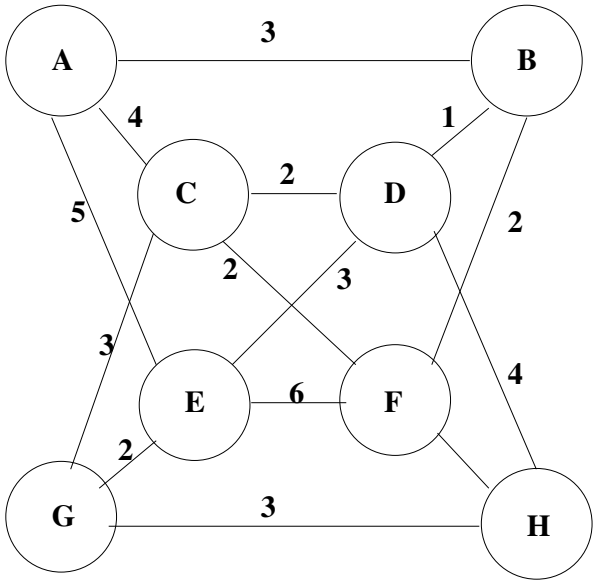
Métodos de Programação II

LESI / LMCC

22 de Julho 2003

4

Questão 4 [*Grafos*]
Considere o seguinte grafo:



1. Classifique o grafo da figura.
2. Desenhe uma árvore de antecessores passível de ser produzida pelo algoritmo de travessia Depth First que estudou nas aulas da disciplina. Considere que o nó A é utilizado como fonte.
3. Escreva a declaração dos tipos de dados adequados para representar este grafo por listas de adjacência, numa implementação mista. Desenhe uma possível representação do grafo na figura utilizando os tipos de dados que definiu.
4. Considere o algoritmo de Prim para o cálculo da Árvore Geradore Mínima de um grafo. Represente o grafo anterior, identificando os conjuntos dos nós da arvore, dos nós da orla e os arcos candidatos ao longo de uma execução do referido algoritmo. Considere que o nó A é o primeiro nó a ser escolhido para ser incluído na árvore.
5. Escreva uma função que receba como parâmetros um grafo e um array de inteiros, e devolva o array preenchido da seguinte forma:
   - Cada componente ligado do grafo o algoritmo fará corresponder um código inteiro sequêncial.
   - O array indicará qual o componente ligado a que pertence o vértice `i`, devendo conter o código inteiro correspondente.

Nome:_____ **Número:**_____ **Curso:**_____