Exame de Qualificação - 07/08/2014
Profs. Álvaro Freitas Moreira e Luciana S. Buriol

Nome:

Dicas gerais:

- Leia todas as questões antes de começar e pergunte em caso de dúvidas;
- sempre justifique a sua resposta;
- responda a cada questão, ainda que a resposta não esteja completa.

Escolha apenas quatro questões para fazer, sendo duas entre as três primeiras e duas das três últimas.

1. (2,5 pontos) **Análise de algoritmos.** Considere o pseudocódigo abaixo. O algoritmo recebe como entrada um nó $s \in V$ e um grafo não direcionado pesado $G = (V, A, w)$ onde V representa o conjunto de nós (considere que estes estejam enumerados de 1 a $|V|$), A representa o conjunto de arcos e $w(u, v) \in R^+$ são valores atribuídos a cada arco $(u, v) \in A$.

```
 1 INPUT: G=(V,A,w), s ∈ V
 2 for each v ∈ V do
 3 |  d[v] := ∞;
 4 end
 5 d[s] := 0;
 6 Q := V;        // Q is initialized with all nodes from V
 7 while Q ≠ ∅ do
 8 |  u := Extract-min(Q);       //node u with the min value of d is removed from Q
 9 |  for each v adjacent to u do
10 |  |  if v ∈ Q and d[v] > w(u,v) then
11 |  |  |  d[v] := w(u, v);
12 |  |  end
13 |  end
14 end
```

Responda as questões abaixo:

a) Qual problema o algoritmo resolve?

b) Dependendo das estruturas de dados utilizadas a complexidade do algoritmo muda. Analise a complexidade de pior caso do algoritmo considerando as seguintes estruturas de dados (considere que $|V|$=n and $|A| = m$):

   i. Q é um heap mínimo binário e G é uma lista de adjacência;

   ii. Q é um heap mínimo binário e G é uma matriz de adjacência.

   iii. Q é um vetor sem organização específica dos nós e G é uma lista de adjacência.

   iv. Q é um vetor sem organização específica dos nós e G é uma matriz de adjacência.

OBS-1: observe que dependendo da estruturas de dados que implementa Q a operação na linha 11 requer uma atualização de Q.

OBS-2: lista de adjacência é a representação de um grafo onde para cada nó é apresentada uma lista dos nós adjacentes. Matriz de adjacência é uma matriz nxn contendo em cada célula [i,j] os valores 1 ou 0 indicando a existência ou não do arco (i,j) em G.

2. (2,5 pontos) Dado um vetor $A[1..n]$ com n valores inteiros. Se um mesmo elemento ocorrer em mais que $\frac{n}{2}$ posições do vetor então este elemento é chamado *maioral*. Assuma que os elementos não podem ser ordenados, mas podem ser comparados. Apresente o pseudocódigo de um algoritmo de divisão e conquista que encontre o elemento maioral em A, ou determine que não existe elemento maioral em A. Analise a complexidade do seu algoritmo.
   Exemplo: O elemento maioral de A=[2 4 2 11 2 8 2 2 2 3] é 2.

3. (2,5 pontos) **Contexto do problema**: Você resolve dirigir 4.300 km e ir de carro visitar um amigo em Fortaleza pela BR-116 (maior rodovia totalmente pavimentada do país). Felizmente o seu celular tem um aplicativo que indica, em qualquer ponto da rodovia, a quantos quilômetros está o próximo posto de gasolina. Considere que:
   - Você decide que sempre que abastecer vai encher o tanque.
   - No tanque do seu carro cabe exatamente 50 litros de gasolina e o seu carro faz exatamente 10km por litro.
   - Considere que no ponto de partida têm um posto de gasolina, e o tanque do seu carro está vazio.

   Você deve decidir em quais postos de gasolina abastecer de forma a minimizar o número de paradas no percurso.

   Problema: Apresente o pseudocódigo de um algoritmo ótimo que dado uma lista de postos (indexados de 1 a n) e para cada posto forneça a distância do próximo posto na rodovia (uma instância exemplo é fornecida abaixo), o algoritmo retorna a lista dos índices dos postos nos quais você vai parar para abastecer. Analise a complexidade de pior caso do seu algoritmo e argumente o porque o seu algoritmo é ótimo.

   $$\boxed{0} \xrightarrow{120} \boxed{1} \xrightarrow{220} \boxed{2} \xrightarrow{40} \boxed{3} \xrightarrow{160} \boxed{4} \xrightarrow{100} \qquad \cdots \rightarrow \boxed{n}$$

4. (2,5 pontos) Disserte sobre a Tese de Church-Turing. Deixe claro (i) o que ela afirma, e (ii) qual o seu significado para a Ciência da Computação. Explique também (iii) porque ela não pode ser provada formalmente e (iv) por que, mesmo sem prova formal da sua verdade, ela é aceita como verdadeira sendo um dos fundamentos da Teoria da Computabilidade.

5. Um dos mais importante e interessantes teoremas da Teoria da Computabilidade é o Teorema de Rice. Ele afirma que toda propriedade não trivial de funções computadas por programas é não decidível. Uma propriedade de funções parciais é dita *trivial* se ela é verdadeira para todas as funções ou para nenhuma. Note que o Teorema se refere a *funções* computadas por programas e não à representação/forma como a função é computada (por exemplo, o número de transições de um programa em uma Máquina de Turing ou o número de linhas de código de um programa em C são propriedades da representação/forma, não da função computada).
   a) Dê pelo menos 3 exemplos de propriedades não triviais de funções computadas por programas
   b) Disserte sobre a significância do Teorema de Rice para a prática da Ciência da Computação

6. (2,5 pontos) Um conjunto S é contável se é vazio ou se existe uma seqüência s_0, s_1, s_2, ... com todos (e somente todos) os elementos de S. Está seqüência é chamada de uma enumeração de S.

   Exemplos de conjuntos contáveis;

   - O conjunto dos números naturais é contável. Uma seqüência óbvia mencionando todos os elementos é $0, 1, 2, \ldots$.
   - O conjunto $\mathbb{Z}$ de todos os números inteiros é contável; e uma seqüência para os seus elementos é $0, 1, -1, 2, -2, 3, -3, \ldots$;
   - O conjunto $\mathbb{N} \times \mathbb{N}$ é contável; uma sequência é:
     $(0, 0), (0, 1), (1, 0), (0, 2), (1, 1), (2, 0), (0, 3), (1, 2), (2, 1), (3, 0), \ldots$

   Esta terminologia também se aplica a conjuntos de funções, parciais ou totais. Sejam A e B conjuntos e seja S um conjunto não vazio cujos elementos são funções parciais de A para B. Então S é contável se existe uma seqüência $f_0, f_1, \ldots$ tal que $g \in S$ se e somente se $g = f_i$ para algum i, ou seja, g ocupa alguma posição i na sequência.

   Prove as seguintes afirmações:

   a) o conjunto de todas as funções totais de $\mathbb{N}$ para $\mathbb{N}$ é incontável;
   b) o conjunto de todas as funções parcias de $\mathbb{N}$ para $\mathbb{N}$ é incontável;
   c) o conjunto de todas as funções computáveis parciais de $\mathbb{N}$ para $\mathbb{N}$ é contável;
   d) o conjunto de todas as funções computáveis totais de $\mathbb{N}$ para $\mathbb{N}$ é contável;
   e) existe função não computável total de $\mathbb{N}$ para $\mathbb{N}$;
   f) existe função não computável parcial de $\mathbb{N}$ para $\mathbb{N}$.

   Observações:

   (I) a prova de (a) acima pode ser feita por contradição usando o método da diagonalização;
   (II) a prova de (b) segue imediatamente do resultado (a) acima e de teoria dos conjuntos;
   (III) a prova de (c) faz uso da Tese de Church-Turing;
   (IV) a prova de (d) é imediata a partir da afirmação (c) e de conhecimentos básicos de funções e de teoria dos conjuntos;
   (V) a prova de (d) e (e) também é imediata a partir dos resultados anteriores e de teoria dos conjuntos.

   Ou seja, o esforço de prova está na verdade nos itens (a) e (c) acima. A prova das demais afirmações pode ser feita assumindo a verdade das afirmações anteriores usando resultados básicos da Teoria dos Conjuntos de acordo com as sugestões acima