MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2001

Prova 3

Nome do aluno: ____

Assinatura: ____

Instruções:

1. Não destaque as folhas deste caderno.
2. Justifique as suas respostas.
3. Preencha o cabeçalho acima.
4. A prova consta de 4 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. A duração da prova é 2:30 horas.

BOA PROVA E BOAS FÉRIAS!!!

Questão	Nota
1	
2	
3	
4	
Total	

1. (Valor: 2,0 pontos) Um **emparelhamento** em um grafo G é um conjunto M de arestas tais que quaisquer duas arestas em M não têm pontas em comum. O seguinte algoritmo guloso produz um emparelhamento em um dado grafo:

   Algoritmo Acha_Emparelhamento (G)

   ```
   1. M := ∅
   2. para cada aresta e em E_G faça
   3.     se M ∪ {e} é um emparelhamento
   4.         então M := M ∪ {e}
   5. devolva M
   ```

   (a) Considerando que o grafo é dado pelas suas listas de adjacências, descreva como implementar o algoritmo de forma que sua complexidade seja $O(m+n)$, onde m é o número de arestas e n o número de vértices do grafo. Explique especialmente que estrutura de dados você usaria e como seriam implementadas as linhas 3 e 4 do algoritmo. Como seria implementado o `para` da linha 2? Argumente que a sua implementação de fato resulta em um algoritmo com complexidade $O(m+n)$.

   (b) Prove ou dê um contra-exemplo para a seguinte afirmação: o algoritmo devolve um emparelhamento de tamanho máximo em G.

2. (Valor: 2,0 pontos)

   (a) Simule o algoritmo de Kruskal no grafo abaixo. Deixe bem claro em que ordem as arestas foram inseridas na árvore.

   (b) Existe uma árvore geradora de custo mínimo no grafo acima contendo a aresta de custo 11? Justifique a sua resposta.

3. (Valor: 2,5 pontos) Um grafo G é **biparticionável** se existe um conjunto X de vértices de G tal que toda aresta de G tem exatamente uma ponta em X. Se você fizer o item (b) certo (ou quase certo), não precisa fazer o item (a). Se você não sabe bem como fazer o item (b), faça pelo menos o item (a) direito.

   (a) Escreva um algoritmo que determine em $O(m+n)$ se um dado grafo conexo é biparticionável, onde m é o número de arestas e n o número de vértices do grafo. A saída do seu algoritmo pode ser apenas "Sim, o grafo é biparticionável" ou "Não, o grafo não é biparticionável". Explique o seu algoritmo e mostre que ele de fato consome tempo $O(m+n)$.

   (b) Escreva um algoritmo que, dado um grafo conexo G, imprime em tempo $O(m+n)$ os vértices de um circuito de comprimento ímpar em G (ou seja, um circuito com um número ímpar de vértices) ou os vértices de um conjunto X de G tal que toda aresta de G tem uma ponta em X e a outra fora de X. Explique o seu algoritmo e mostre que ele de fato consome tempo $O(m+n)$.

4. (Valor: 3,0 pontos)

   (a) Defina precisamente as classes P e NP de problemas.

   (b) O que significa dizer que um problema Q é NP-completo? (Não dê a definição de NP-completude; diga o que você entende por isso.)

   (c) Marque F, V e O para, respectivamente, *falso*, *verdadeiro* e *não se sabe.*

   (   ) P $\subseteq$ NP

   (   ) P $\neq$ NP

   (   ) Existem problemas NP-completos em P.

   (d) Neste item, usaremos a mesma notação usada com o problema SAT, visto em aula. Em particular, consideramos apenas fórmulas booleanas envolvendo variáveis, suas negações e os operadores $\vee$ (ou) e $\wedge$ (e). O seguinte problema é conhecido como TAUTOLOGIA: dada uma fórmula booleana Φ, com m cláusulas sobre as variáveis $x_1, \ldots, x_n$, determinar se todas as valorações de $x_1, \ldots, x_n$ satisfazem Φ. (Se facilitar, suponha que a fórmula é dada em forma normal conjuntiva, como no SAT.)

   Prove que TAUTOLOGIA está em co-NP.