MC538 - Análise de Algoritmos II
MC548 - Projeto e Análise de Algoritmos II
Semestre 2003-1 - Turma A
Exame Final - 01/jul/2003

Nome	
RA	Assinatura
Notas	

A prova é individual e sem consulta.
Não são permitidos computadores ou calculadoras.
Não separe as folhas deste caderno de prova.
Não é permitido o uso de outro rascunho além destas folhas.
Escreva seu nome completo, e assine a tinta.
Valem apenas as respostas nos espaços indicados.

Nas questões tipo teste:
- **marque apenas uma das cinco alternativas (a)-(e).**
- **marcar mais de uma alternativa conta como erro.**
- **cada 4 questões erradas anulam uma certa.**
- **questões em branco não contam nem descontam.**

1. Suponha que p e r são predicados, e r é uma restrição de p. Isto é, $r(x)$ é igual a $p(x)$ para todo x pertencente a um subconjunto D das possíveis entradas de p, e não está definido se x não pertence a D. Supondo $\mathrm{P} \neq \mathrm{NP}$, qual das seguintes afirmações é sempre verdadeira:

   (a) ☐ Se p é NP-completo, então r é NP-completo.
   (b) ☐ Se p é NP-difícil, então r é NP-difícil.
   (c) ☐ Se r está em P, então p está em NP.
   (d) ☐ Se r é NP-completo, então p é NP-completo.
   (e) ☐ Se r é NP-completo, então p é NP-difícil.

2. Sabe-se que o problema p está em P, e o problema q é NP-completo. Qual das afirmações abaixo é portanto **falsa**:

   (a) ☐ Existe um problema NP-difícil r que pode ser reduzido a q em tempo polinomial.
   (b) ☐ Existe uma redução de p para q que leva tempo polinomial.
   (c) ☐ Qualquer problema r que pode ser reduzido a q em tempo polinomial está em NP.
   (d) ☐ Existe uma redução de q para p que toma tempo polinomial.
   (e) ☐ Se for encontrado um algoritmo polinomial para q, estará provado que $\mathrm{P} = \mathrm{NP}$.

3. Um predicado p pode ser escrito composição de duas outras funções, $p(x) = q(h(x))$:

$$
\begin{tikzcd}
x \arrow[r, "h"] \arrow[d, "p"'] & y \arrow[d, "q"] \\
V/F & V/F \arrow[l]
\end{tikzcd}
\tag{1}
$$

   onde a seta inferior representa a função identidade. Suponha que a função h pode ser calculada em tempo polinomial em $n = |x|$. Podemos concluir que:

   (a) ☐ Se q é NP-completo então p é NP-completo.
   (b) ☐ Se p é NP-difícil, então q é NP-completo.
   (c) ☐ Se p é NP-completo, então q é NP-difícil.
   (d) ☐ Se q é NP-difícil, então p é NP-difícil.
   (e) ☐ Se p está em NP, então q está em NP.

4. Considere o predicado `ISO` que, dado um par de grafos, devolve **verdadeiro** se e somente se os grafos são isomorfos. O que se sabe atualmente sobre a classe de complexidade (P, NP, etc.) desse problema?

   (a) ☐ Não se sabe se ele está em NP.
   (b) ☐ Sabe-se apenas que ele está em NP.
   (c) ☐ Sabe-se apenas que ele é NP-difícil.
   (d) ☐ Sabe-se que ele não está em P
   (e) ☐ Sabe-se que ele é NP-completo.

5. Considere o problema da mochila binária: dados inteiros $V_{\min}$ e $T_{\max}$, e n itens com valores $v_1, v_2, \ldots, v_n$ e tamanhos (pesos) $t_1, t_2, \ldots, t_n$, encontrar um subconjunto X de $N = \{1, 2, \ldots, n\}$ tal que $\sum_{i \in X} v_i \geq V_{\min}$ e $\sum_{i \in X} t_i \leq T_{\max}$. Na heurística construtiva "gulosa" para este problema, vista em classe, os itens são considerados:

   (a) ☐ Em ordem decrescente da razão t_i/v_i.
   (b) ☐ Em ordem crescente da razão t_i/v_i.
   (c) ☐ Em ordem crescente da diferença $v_i - t_i$.
   (d) ☐ Em ordem crescente de tamanho t_i.
   (e) ☐ Em ordem crescente de valor v_i.

6. No problema de empacotamento de arquivos em volumes, deseja-se copiar n arquivos de tamanhos $t_1, t_2, \ldots, t_n$ para m volumes, cada um com capacidade L. A função objetivo, a ser maximizada, é o **número** de arquivos gravados. Considere o algoritmo visto em classe, para 2 volumes, e generalize-o para um número arbitrário m de volumes. Se Z_* é o valor da solução ótima, então o algoritmo generalizado A consegue uma aproximação Z_A tal que:

   (a) ☐ $Z_*/Z_A \leq 2$.
   (b) ☐ $Z_*/Z_A \leq 2 - 1/m$.
   (c) ☐ $|Z_* - Z_A| \leq m - 1$.
   (d) ☐ $|Z_* - Z_A| \leq 1$.
   (e) ☐ $|Z_* - Z_A| \leq \lfloor m/2 \rfloor$.

7. Nesta questão você deve projetar um algoritmo do tipo **divide-e-limita** (*branch-and-bound*) para o seguinte problema. São dados: um conjunto de itens identificados pelos índices $N = \{1, .. n\}$, um custo positivo $c_{i,j}$ associado a cada par de itens $(i, j) \in N \times N$, e um inteiro m entre 1 e n. Deseja-se encontrar um subconjunto X de N, **com exatamente** m **elementos**, que **minimiza** a soma $S(X)$ dos custos de todos os pares de elementos escolhidos, isto é $S(X) = \sum_{i=1}^{m} \sum_{j=i}^{m} c_{i,j}$.

   (a) Descreva a representação do estado genérico (solução parcial), e o estado inicial.

   (b) Defina os estados "filhos" de um estado genérico.

   (c) Defina os estados-folha.

   (d) Dê uma função limitante inferior adequada para cada estado.

   (e) Descreva o critério para 'podar' a sub-arvore dos descendentes um nó.

8. Encontre todas as soluções da equação $56x \equiv 35 \pmod{91}$.

9. Seja $n = p^2 \cdot q$, onde p e q são dois primos distintos. Quais são os elementos de $\mathbb{Z}_n^*$?

10. Calcule $7^{22} \bmod 50$ pelo algoritmo de exponenciação rápida. (Mostre as contas.)

11. A questão abaixo se refere ao algoritmo RSA para ciframento com chave pública. Para facilitar as contas, vamos usar um módulo n de apenas **quatro** algarismos.

    (a) Escolha valores adequados dos fatores p e q, entre 33 e 99:

    (b) Escolha uma chave pública adequada.

    (c) Determine a chave secreta correspondente à escolha acima.