Universidade Estadual de Maringá
Departamento de Informática – Bacharelado em Informática
5184 – Projeto e Análise de Algoritmos / Prof. Daniel Kikuti

Aluno(a): ____

Segunda avaliação (Valor: 10,0)

1. [Valor: 2,0] Marque (V)erdadeiro ou (F)also[1].
   (a) ☐ V ☐ F O problema de verificar se um número representado em binário é par está em P .
   (b) ☐ V ☐ F Se P $\neq$ NP então nenhum problema NP pode ser resolvido em tempo polinomial.
   (c) ☐ V ☐ F Se P $\neq$ NP então nenhum problema NP-Completo pode ser resolvido em tempo polinomial.
   (d) ☐ V ☐ F Se $X \in NP$ e existe um algoritmo polinomial que resolve X, então P = NP.
   (e) ☐ V ☐ F Se $P = NP$, então nenhum problema demanda tempo exponencial para ser resolvido.
   (f) ☐ V ☐ F É correto afirmar que 2-SAT pertence às classes P e NP.
   (g) ☐ V ☐ F Se há um algoritmo que resolve 3-SAT com complexidade $O(n^k)$, onde n é o número de literais e k é uma constante, então P = NP.
   (h) ☐ V ☐ F Se um problema X é NP-Completo, então existe um algoritmo de tempo polinomial não-determinístico que resolve X.
   (i) ☐ V ☐ F Suponha que X é NP-Completo e existe um algoritmo de tempo polinomial que resolve Y. Se $P \neq NP$, então é possível efetuarmos a redução $X \leq_p Y$, mas não a redução $Y \leq_p X$.
   (j) ☐ V ☐ F Suponha que X é NP-Completo e existe um algoritmo de tempo polinomial que resolve Y. Se $P \neq NP$, então é possível efetuarmos tanto a redução $X \leq_p Y$, quanto a redução $Y \leq_p X$.

2. [Valor: 2,0] Explique o que é redução em tempo polinomial. Apresente um exemplo de como podemos usá-la para mostrar que um problema X é NP-Completo (preferencialmente use o exemplo do seu trabalho).

3. [Valor: 3,5] Considere o seguinte problema de seleção de atividades. "Dado um conjunto de n atividades $S = \{a_1, \ldots, a_n\}$ que requerem o uso de um recurso comum e, os tempos de início e término de cada atividade $[s_i, f_i)$, selecionar o maior conjunto possível de atividades mutuamente compatíveis, isto é, atividades a_i e a_j tais que $s_i \geq f_j$ ou $s_j \geq f_i$."
   (a) [Valor: 1,0] Seja S_{ij} o conjunto de atividades que começam após o término da atividade a_i e terminam antes do início da atividade a_j. Denote por A_{ij} um conjunto máximo de atividades mutuamente compatíveis em S_{ij} que contém alguma atividade a_k. Mostre que esta formulação para o problema apresenta subestrutura ótima.
   (b) [Valor: 0,8] Seja $|A_{ij}|$ o tamanho de uma solução ótima para o conjunto S_{ij}. Considerando a recorrência a seguir, apresente um algoritmo que faz uso da técnica de programação dinâmica para resolver este problema.

$$|A_{ij}| = \begin{cases} 0 & \text{se } S_{ij} = \emptyset \\ \max\limits_{a_k \in S_{ij}} (|A_{ik}| + |A_{kj}| + 1) & \text{se } S_{ij} \neq \emptyset \end{cases}$$

   (c) [Valor: 0,7] Analise a complexidade do algoritmo desenvolvido no item anterior no pior caso.
   (d) [Valor: 1,0] O algoritmo a seguir resolve o problema corretamente? Justifique.

   Repita enquanto $S \neq \emptyset$
      Selecione a_k cujo intervalo sobrepõe menos as demais atividades
      Adicione a_k no conjunto S'
      Remova todos as atividades de S que sobrepõe a_k

4. [Valor: 1,5] Considere o problema de Soma de Subconjuntos descrito a seguir: "Dado uma coleção de números naturais $S = \{a_1, \ldots, a_n\}$ e um valor s, existe um subconjunto S' de S tal que $\sum_{x \in S'} x = s$?" Apresente um algoritmo guloso para este problema. Mostre que ele está correto ou apresente uma instância em que ele falha.

Boa Prova!!!

[1] Quando não estiver explicitamente especificado em qual máquina, considere máquina determinística.

Aluno(a): ______________________________