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 x pertence a um conjunto de n números 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 O problema de verificar se uma fórmula booleana é satisfazível pertencente à classe NP.
   (e) ☐ V ☐ F Se há um algoritmo que resolve o problema de *Subset Sum* em $O(n^{100})$, então P = NP.
   (f) ☐ V ☐ F Suponha que $X \in$ NP e existe um algoritmo que resolve X em $O(\lg n)$, então P = NP.
   (g) ☐ V ☐ F Se $P = NP$, então nenhum problema demanda tempo exponencial para ser resolvido.
   (h) ☐ V ☐ F Se um problema X é NP-Completo, então existe um algoritmo de tempo polinomial não-determinístico que resolve X.

2. [Valor: 1,0] Sabe-se que 3-SAT é NP-Completo e que existe um algoritmo de tempo polinomial para o problema 2-SAT. Se assumirmos que P $\neq$ NP, então é possível termos 3-SAT $\leq_p$ 2-SAT? Justifique.

3. [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).

4. [Valor: 2,5] 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] Mostre que o problema possui subestrutura ótima.
   (b) [Valor: 1,5] Apresente um algoritmo guloso para o problema e argumente se o algoritmo apresentado está correto ou não.

5. [Valor: 2,5] Considere o problema de corte de hastes definido a seguir. "Dada uma haste de tamanho n e uma tabela de preços p_i, para $1 \leq i \leq n$, determinar a receita máxima r_n obtida após cortar a haste e vender os pedaços." O algoritmo a seguir faz uso da subestrutura ótima do problema (assumindo que $r_0 = 0$), isto é, $r_n = \max_{1 \leq i \leq n}(p_i + r_{n-i})$.

```
Cut-Rod(p,n)
if n == 0
   return 0
q = −∞
for i = 1 to n
   q = max(q, p[i] + Cut-Rod(p,n-i))
return q
```

   (a) [Valor: 1,5] Agora, considere uma modificação do problema de corte de hastes no qual, além de um preço p_i cada corte incorre em um custo fixo c. A receita associada à solução agora é a soma dos preços das peças menos os custos da execução dos cortes. Escreva um algoritmo que usa a técnica de programação dinâmica para este novo problema.
   (b) [Valor: 1,0] Mostre que a seguinte estratégia gulosa nem sempre determina uma maneira ótima de se fazer cortes em hastes. Defina a *densidade* de uma haste de comprimento i como sendo p_i/i, isto é, seu valor por polegadas. A estratégia gulosa (para uma haste de tamanho n) corta um pedaço de tamanho i $(1 \leq i \leq n)$, com a densidade máxima. Então, continua-se aplicando a estratégia gulosa no pedaço restante de tamanho $n - i$.

Boa Prova!!!

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

Aluno(a): ______________________________