Aluno(a): ______________________________

Terceira avaliação (Valor: 10,0)

1. [Valor: 2,0] Marque (V)erdadeiro ou (F)also.
   (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 de tempo $O(n^{100})$ para o problema de *Subset Sum*, então P = NP.
   (f) ☐ V ☐ F Suponha que $X \in$ NP. Se existir um algoritmo de tempo $O(\lg n)$ para X, então P = NP.
   (g) ☐ V ☐ F O problema de parada da Máquina de Turing é NP-Completo.
   (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] Um problema X é NP-Difícil se e somente se existe um problema NP-Completo Y tal que Y $\leq_p$ X. X não precisa estar em NP e nem ser um problemas de decisão. Seja S um problema NP-Completo, Q e R problemas que não sabemos se pertencem a NP. Se existe uma redução em tempo polinomial de Q para S e, S pode ser reduzido em tempo polinomial para R, assinale qual(is) sentença(s) é(são) verdadeira(s)?

   ☐ R é NP-Completo ☐ R é NP-Difícil ☐ Q é NP-Completo ☐ Q é NP-Difícil

3. [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.

4. [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.

5. [Valor: 1,0] Um algoritmo verificador para um problema de decisão recebe dois objetos: uma instância do problema e um certificado. Ao receber esses dois objetos, o verificador pode responder SIM ou NÃO. Se responder SIM, dizemos que o verificador aceitou o certificado. Um verificador para um determinado problema de decisão é polinomial se: (i) para cada instância positiva do problema, existe um certificado que o verificador aceita em tempo limitado por uma função polinomial do tamanho da instância; (ii) para cada instância negativa do problema, não existe certificado que o verificador aceite. Dada esta definição, apresente um algoritmo verificador para o Problema de Cobertura de Vértice. [Dado um grafo $G = (V, E)$, uma cobertura de vértices é um subconjunto de vértices $S \subseteq V$ tal que toda aresta $(u, v) \in E$ tem pelo menos uma ponta (u ou v) em S.]

6. [Valor: 1,5] Um clique em um grafo $G = (V, E)$ é um subgrafo completo de G (cada dois vértices do subconjunto são conectados por uma aresta). Encontrar um clique de tamanho máximo em um grafo é um problema de otimização para o qual não se conhece um algoritmo que o resolva em tempo polinomial. Mas se P=NP, então mostre que existe um algoritmo que resolve o Problema do Clique Máximo em tempo polinomial. [Note que, o problema de verificar se um grafo possui um clique de tamanho $\geq k$ é NP-Completo. Assuma que CLIQUE(G,k) é o algoritmo que resolve este problema. Dica: desenvolva seu algoritmo de modo a fazer diversas consultas a CLIQUE(G,k).]

7. [Valor: 1,5] Um conjunto independente de um grafo $G = (V, E)$ é um conjunto de vértices $S \subseteq V$ tal que não existem dois vértices adjacentes contidos em S. Vimos que o problema de encontrar um conjunto independente de tamanho k em um grafo é um problema NP-Completo. E o problema de encontrar em um grafo um conjunto independente contendo exatamente 3 vértices? Também é um problema NP-Completo? Justifique.

Boa Prova!!!

Aluno(a): ______________________________