MC558: Análise de Algoritmos II
Lista de Classes de Complexidade – Parte 2

1. Dois problemas P e Q são **polinomialmente equivalentes** se $P \propto_{\text{poli}} Q$ e $Q \propto_{\text{poli}} P$. Usando a definição de $\mathcal{NP}$-completude, mostre que todos problemas em $\mathcal{NP}$-completo são polinomialmente equivalentes.

2. Dê os argumentos que mostram que todos os problemas em $\mathcal{P}$ são polinomialmente equivalentes.

3. Faça um diagrama de Venn que represente a situação atual do conhecimento que se tem sobre as classes de problemas $\mathcal{NP}$, $\mathcal{P}$, $\mathcal{NP}$-completo e $\mathcal{NP}$-difícil.

4. Mostre que a função de redução $\propto_{\text{poli}}$ é transitiva.

5. Escreva o problema da árvore geradora mínima em suas versões de decisão e de otimização.

6. Escreva o problema da cobertura de vértices mínima em suas versões de decisão e de otimização.

7. Escreva qual é o problema complementar ao problema da partição (`PAR`) que foi dado em sala de aula.

8. Defina a classe de problemas co-$\mathcal{NP}$.

9. Mostre que $\mathcal{P}=$ co-$\mathcal{P}$.

10. Defina as classes de problemas $\mathcal{PSPACE}$ e $\mathcal{NPSPACE}$.

11. Mostre que $\mathcal{P} \subseteq$ co-$\mathcal{NP} \cap \mathcal{NP}$.

12. Desenhe um diagrama de Venn que represente o conhecimento atual sobre a relação existente entre as classes $\mathcal{NP}$, $\mathcal{P}$ e co-$\mathcal{NP}$.

13. Suponha que você está estudando um certo problema π. Você já conseguiu provar que o problema $\overline{\pi}$ (i.e., o problema complementar de π) está em $\mathcal{NP}$.

    Se você tivesse que investir seus esforços em uma das alternativas abaixo, por qual delas você optaria? Justifique a sua resposta.

    Alternativa 1: encontrar um algoritmo determinístico polinomial para π.

    Alternativa 2: provar que π é $\mathcal{NP}$-completo.

14. Suponha que você está estudando um problema π e que você encontrou um algoritmo **<u>não determinístico</u>** cuja complexidade de espaço é n^2. Um amigo seu disse ter "provado" que nenhum algoritmo **<u>determinístico</u>** pode resolver este problema usando menos que $O(n^5)$ de memória. Há alguma contradição entre os dois resultados? Justifique a sua resposta.

15. Escreva em um pseudo-código de alto nível um algoritmo **não-determinístico polinomial** que resolve o problema do conjunto dominante visto em aula. Ou seja, mostre que este problema está em $\mathcal{NP}$. Qual a complexidade do seu algoritmo?

16. Considere o problema do empacotamento descrito abaixo e doravante denotado por `BIN`.

    Instância: Um conjunto finito de n objetos com pesos $w_1, w_2, \ldots, w_n$ inteiros positivos. Dois valores inteiros positivos W e k.

    Questão: É possível colocar todos os objetos em k caixas cujo limite máximo de peso é W?

    Escreva em um pseudo-código de alto nível um algoritmo **não-determinístico polinomial** que resolve `BIN`. Ou seja, mostre que este problema está em $\mathcal{NP}$. Qual a complexidade do seu algoritmo?

17. Considere os problemas P_1 de $\mathcal{P}$ e P_2 de $\mathcal{NP}$-completo. Indique para cada uma das afirmações abaixo se ela é **verdadeira**, **falsa** ou se **não se sabe**.

    (a) Existe uma redução polinomial de P_1 para P_2.
    - Existe uma redução polinomial d eP_2 para P_1.
    - Se existe um algoritmo determinístico polinomial para resolver P_2 então $\mathcal{P} = \mathcal{NP}$.
    - Se P_3 é um problema $\mathcal{NP}$-difícil, então P_3 pode ser reduzido polinomialmente a P_2.
    - Se P_4 é um problema em $\mathcal{P}$, então P_4 não pode ser reduzido polinomialmente a P_1.
    - Se P_4 é um problema em $\mathcal{P}$ e P_4 pode ser reduzido polinomialmente a P_2, então $\mathcal{P} = \mathcal{NP}$.
    - Pode-se afirmar que P_1 está em $\mathcal{NP}$-espaço.
    - Pode-se afirmar que P_2 está em $\mathcal{P}$-espaço.