



ALUNO(A): FRANCISCO PIRES JÚNIOR NOTA: _____

ATENÇÃO: Descrever as soluções com o máximo de detalhes possível, no caso de programas (escritos em C ou C++), inclusive a forma como os testes foram feitos. Todos os artefatos (relatório, código fonte de programas, e outros) gerados para este trabalho devem ser adicionados em um repositório no site `github.com` com os seus pontos extras da disciplina.

1) Descreva o que é a NP-Compleitude.

Existem alguns problemas computacionais que são difíceis de serem resolvidos, sendo impossíveis de se provar que não existem soluções eficientes. Um algoritmo é dito eficiente quando a sua complexidade for polinomial em relação ao tamanho de sua entrada. Basicamente, um algoritmo é dito ser de tempo polinomial se for $O(n^k)$, para alguma constante $k > 0$. Qualquer outro algoritmo que não for polinomial é dito ser exponencial. Um problema é dito **tratável** se ele apresenta uma solução polinomial e um problema é intratável se ele for tão difícil que nenhum algoritmo polinomial pode resolvê-lo. Essa é ideia central da teoria de **NP-Compleitude**.

2) Apresente 5 problemas provados ser NP-Completo, com suas respectivas referências.

- **Problema de Satisfatibilidade, disponível em** <https://www.youtube.com/watch?v=wsvXnmUHJX8>
- **Equação do segundo grau, disponível em** https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html
- **Fatoração, disponível em** https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html
- **Problema do ciclo longo, disponível em** https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html
- **Problema do ciclo hamiltoniano, disponível em** https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto.html

3) Apresente um lauda sobre o artigo:

S. Cook, *The complexity of theorem-proving procedures*, Proceedings of the 3rd Symposium on the Theory of Computing, ACM, pp.151-158. 1971.

Disponível em: <https://dl.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=805047>

