

Trabalho Prático – Algoritmos e Grafos
Caminho Hamiltoniano: Complexidade, Algoritmos e Experimentos
Especificação – 21/11/2025
Entrega - 10/01/2026

1 Objetivo geral

O objetivo deste trabalho é explorar um problema clássico de algoritmos envolvendo caminhos em grafos, cuja versão de decisão está em NP-completo. Espera-se que o estudante possa demonstrar as seguintes habilidades:

- analisar formalmente a complexidade via reduções polinomiais;
- realizar uma pesquisa bibliográfica sobre possíveis soluções existentes, contendo algoritmos exatos, heurísticas ou aproximações;
- entregar implementação prática de soluções;
- realizar experimentos e analisar seus resultados, contemplando instâncias de diferentes tamanhos;
- elaborar um relatório técnico contendo todos estes itens acima mencionados.

2 Caminho Hamiltoniano

O problema do Caminho Hamiltoniano é definido como segue:

Entrada: Um grafo $G = (V, E)$.

Problema de decisão: Existe um caminho simples que visita todos os vértices de G exatamente uma vez?

Saída: **SIM** se um tal caminho existe; **NÃO** caso contrário.

3 Tarefas do trabalho

Parte A – Prova da Complexidade (redução polinomial)

Você deverá selecionar um problema NP-completo conhecido (por exemplo: SAT, Clique, Ciclo Hamiltoniano, Cobertura de Vértices); apresentar uma redução polinomial do problema escolhido para o problema do Caminho Hamiltoniano; apresentar a redução de forma clara, com justificativas passo a passo; e fornecer um exemplo ilustrativo pequeno que acompanhe a redução e sua correção. A ilustração não deve ser tomada como demonstração e sim servir como forma de apresentar a intuição por em torno da demonstração.

Parte B – Pesquisa e seleção de algoritmos

Você deverá pesquisar diferentes abordagens para resolver o problema alvo, incluindo algoritmos exatos (backtracking, branch-and-bound, programação dinâmica, redução para SAT/ILP) assim como heurísticas e métodos aproximados. Em seguida, após contextualizar estas diferentes abordagens estudadas, você deverá selecionar pelo menos duas, descrevendo os princípios de funcionamento, complexidades, vantagens e limitações.

Parte C – Implementação

A implementação deverá ler um grafo a partir de um arquivo de entrada, em formato definido a seguir. Para o grafo lido, é necessário resolver o problema de decisão, retornando SIM ou NÃO. Diferentes tamanhos de instâncias devem ser contemplados. A linguagem de programação a ser usada pode ser Python, C ou C++.

A primeira linha do arquivo de entrada é contém valores inteiros de n e m , representando o número de vértices e de arestas, respectivamente. Cada uma das m linhas seguintes contém dois inteiros v e u , representando uma aresta não-direcionada (v, u) . Como ilustração, considere o seguinte arquivo de entrada:

```
10 15
8 9
0 1
0 4
1 2
1 3
2 5
2 6
3 7
4 5
6 7
```

Tal entrada representa um grafo com $n = 10$ vértices e $m = 15$ arestas. Note que o vértice 0 se conecta com os vértices 1 e 4 enquanto que o vértice 2 está conectado com os vértices 1, 5 e 6.

Parte D – Experimentos

Uma análise experimental deve contemplar o desempenho das soluções escolhidas para diferentes grafos de entrada. Para tanto, deve haver a criação de grafos de diferentes tamanhos (por exemplo, com 10, 20, 30, 40, 50 vértices). Seria interessante que a criação destes grafos fosse aleatória, incluindo conjuntos com grafos esparsos e densos. Para comparar as soluções implementadas, defina métricas, tais como tempo de execução, contagem de interações e/ou memória consumida, o que julgar mais apropriado para sua implementação, justificando suas escolhas. Por fim, de posse dos resultados experimentais, apresente-os em forma de gráficos e tabelas, comentando os resultados encontrados.

Parte E – Relatório

O relatório deverá conter **introdução** ao problema, contexto teórico e relevância; **Parte teórica**, incluindo explicação detalhada da redução polinomial escolhida; **Descrição** dos algoritmos implementados; **Descrição das instâncias**: como foram geradas e suas propriedades; **Resultados experimentais**: tabelas, gráficos e análise crítica; e **Conclusões**. O link para o repositório de código (GitHub, GitLab etc.) será necessária para possíveis verificações de correção da implementação.

4 Critérios de avaliação

Serão observados cinco critérios, todos com mesmos pesos: (a) redução polinomial; (b) pesquisa e descrição dos algoritmos existentes; (c) implementação; (d) experimentos e análise; (e) qualidade do relatório. A segunda nota do semestre será a média aritmética entre a nota deste trabalho e a da segunda prova.