



O trabalho deverá ser feita em dupla ou trio. Ele deverá ser entregue no Canvas até às 23:59 horas do dia 15/06/2022 e valerá 15 pontos. Cópias serão sumariamente zeradas.

Você deverá entregar além dos códigos implementados, um relatório (em formato PDF e também os fontes em TeX) descrevendo detalhes da implementação, dos experimentos e resultados obtidos, além da descrição do solicitado no enunciado do trabalho. Indique as responsabilidades e o que foi feito por cada membro do grupo. O trabalho será avaliado considerando a qualidade do código, a qualidade do texto e a correção da solução entregue.

## QUESTION

Algoritmos baseados em grafos são usados em diversas áreas para auxiliar nas resoluções de inúmeros problemas. Considere  $G = (V, E)$  um grafo, em que  $V$  representa o conjunto de vértices e  $E$  o conjunto de arestas. A Figura 1 ilustra um grafo não-direcionado simples (isto é, sem *loops* e arestas paralelas).

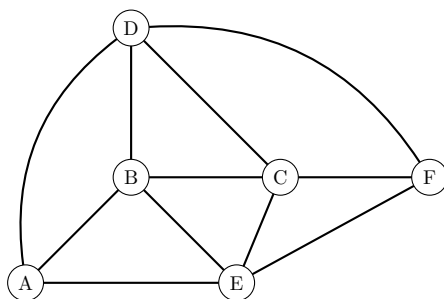


Figura 1: Exemplo de grafo com 6 vértices e 11 arestas.

Um ciclo em um grafo não-direcionado simples é um caminho fechado sem vértices repetidos. Mais precisamente, um ciclo é uma sequência  $(v_0, a_1, v_1, a_2, v_2, \dots, a_k, v_k)$  com  $k > 2$  em que  $v_k = v_0$  mas  $v_0, v_1, v_2, \dots, v_{k-1}$  são distintos dois a dois. Em grafos simples, pode-se representar um ciclo apenas pela sequência de vértices (uma vez que só pode existir uma única aresta entre cada par de vértices). No grafo ilustrado na Figura 1 as sequências  $(A, B, D, C, E, A)$  e  $(C, F, E, C)$  são exemplos de ciclos.

O problema de se **enumerar todos os ciclos** existentes em um grafo apresenta várias aplicações e pode ser resolvido por diferentes abordagens. Neste trabalho você deverá implementar e comparar duas formas distintas de resolução deste problema: (i) uma baseada na permutação dos vértices do grafo; e (ii) outra baseada em caminhamento no grafo.

Além das implementações, você deverá realizar uma análise comparativa entre elas, mostrando o custo computacional teórico (em termos de notação  $O$ ) e prático (em termos de segundos), visando determinar diferenças no desempenho das mesmas para resolução do problema, principalmente na medida que o tamanho do grafo aumentar.

## QUESTION

*Branch and Bound* e *Backtracking* são duas estratégias no desenvolvimento de algoritmos. Discorra sobre as duas estratégias indicando, pelo menos, os conceitos, quando se deve utilizar, além de mostrar exemplos (com implementação). Apresente também as principais semelhanças e diferenças entre estas estratégias. Ainda, se houver semelhanças ou diferenças com as estratégias estudadas na disciplina, a saber, Abordagem Gulosa, Programação Dinâmica e Divisão-e-Conquista, apresente-as.