

**Aluna:** Ana Cláudia Machado

**Entrada:**

A entrada utilizada é um arquivo .txt, intitulado "graph", contendo as arestas. Assim, em cada linha do arquivo, têm-se uma aresta do grafo de entrada, de modo que o primeiro vértice é o de saída e o segundo, conseqüentemente, o de entrada. Utiliza-se a biblioteca networkx para realizar a leitura das arestas e, em seguida, utiliza-se listas para armazenar e trabalhar com os vértices e arestas.

**Plotagem do grafo:**

A biblioteca citada também é utilizada na plotagem do grafo, de modo a facilitar a visualização das componentes fortemente conexas (CFC) existentes.

**Funcionamento do algoritmo:**

A princípio, têm-se as listas para os fechos transitivos direto ( $r\_plus$ ) e inverso ( $r\_minus$ ), os quais iniciam contendo apenas o vértice  $v$  em relação ao qual os fechos serão definidos. O algoritmo inicia com  $v$  sendo o primeiro vértice da lista "vertices". Para definir os fechos transitivos direto e inverso, utilizam-se funções com funcionamentos semelhantes, haja vista que a lógica para selecionar os vértices que pertencem a tais fechos é a mesma. Nesse sentido, em ambos os casos utiliza-se dois *for* aninhados: o externo roda para todos os vértices no fecho e o interno, para todos os vértices no grafo, de modo que:

- Para cada vértice no fecho, procura-se todos os vértices que ele atinge, no caso do fecho transitivo direto, e todos os vértices que o atingem, no caso do fecho transitivo inverso;
- Esses vértices são adicionados ao fecho transitivo, de modo que o *for* externo rodará para eles também.

Assim, define-se os fechos transitivos direto e inverso de  $v$ . O próximo passo do algoritmo é verificar quais dos vértices do grafo estão tanto em  $r\_plus$  quanto em  $r\_minus$ , pois a intersecção desses dois conjuntos de vértices é uma CFC do grafo  $G$ . A partir daí, retira-se os vértices da CFC de  $G$  e repete-se o processo até que  $G$  não tenha mais nenhum vértice.

Para armazenar as CFCs é utilizada uma lista de listas (*result*), a qual é atualizada toda vez que uma CFC é descoberta.

**Saída:**

A saída possui três formas de visualização: no terminal, em um arquivo .txt (*comp-f-conexas*) e em uma imagem .jpg (*output-graph*). Dessa forma, é possível visualizar o resultado da forma que mais agrada ao usuário. Nas saídas de texto, também é mostrado o número de CFCs. Já na saída visual, cada CFC possui uma cor, de modo que fica mais prático compreender a saída.

Para gerar esse grafo de saída, utiliza-se uma lista (*color\_map*) para armazenar as cores geradas aleatoriamente para cada CFC. Para isso, utiliza-se o código RGB e, a cada nova cor gerada pela combinação de 6 elementos do conjunto "0123456789ABCDEF", verifica-se se ela já não faz parte da lista de cores que compõem os vértices do grafo de saída. Se sim, gera-se novamente até que se tenha uma cor inédita.

A cor gerada é adicionada a *color\_map* de acordo com a quantidade de vértices que aquela determinada CFC possui no grafo de saída. Por fim, basta utilizar essa lista ao desenhar o grafo e salvá-lo no arquivo .jpg.