

Atenção: Crie um arquivo-fonte em C para cada questão da lista. Os arquivos-fonte em C que compõem a solução para as questões da lista devem ser salvos em uma pasta com o nome do aluno, compactados e entregues a professora, que os recolherá através de *pen-drive* no dia da prova.

Questão 1: Verificar se grafo é acíclico (5 pontos)

Arestas de retorno são arestas (u,v) conectando um vértice u com um antecessor v em uma **árvore de busca em profundidade**. As arestas *self-loops* são consideradas arestas de retorno. Se uma aresta de retorno é encontrada em uma busca em profundidade em um grafo $G = (V,A)$, então o grafo tem ciclo. Assim, um grafo direcionado é **acíclico**, se e somente se, a busca em profundidade em G não apresentar arestas de retorno. O algoritmo busca em profundidade pode ser alterado para descobrir arestas de retorno. Basta verificar se um vértice v adjacente a um vértice u apresenta a cor cinza na primeira vez que a aresta (u,v) é percorrida. Isso deve ser feito no momento que a lista de adjacentes está sendo percorrida. Implemente essa modificação no algoritmo busca em profundidade para verificar se um grafo é acíclico.

Questão 2: Ordenação topológica (5 pontos)

A ordenação topológica de um grafo direcionado acíclico $G = (V,A)$ é uma ordenação linear de todos os seus vértices tal que se G contém uma aresta (u,v) então u aparece antes de v . É uma ordenação dos vértices ao longo de uma linha horizontal de tal forma que todas as arestas estão direcionadas da esquerda para a direita. Esse tipo de ordenação é útil para representar precedência entre eventos. A ordenação topológica do grafo fornece a ordem que as atividades devem ser processadas baseada nos tempos de descoberta e de término da busca em profundidade. O pseudocódigo a seguir apresenta o algoritmo para ordenar topologicamente um grafo direcionado acíclico $G = (V,A)$:

1. Chama busca em profundidade(G) para obter os tempos de término $t[u]$ para cada vértice u .
2. Ao término de cada vértice, insira-o na frente de uma lista linear encadeada.
3. Retorna a lista encadeada de vértices.

Modifique a função de busca em profundidade para realizar a ordenação topológica de um grafo. O local para realizar a inserção na lista é logo após o momento em que o término $t[u]$ é obtido e o vértice é pintado de preto. Basta retornar a lista obtida (ou imprimi-la). A Figura 1 mostra um exemplo de grafo direcionado acíclico antes e após a ordenação topológica.

Questão 3: Caminho mais curto (5 pontos)

A função de busca em largura obtém a distância do vértice de origem $u \in V$ para qualquer vértice alcançável $v \in V$ em um grafo $G = (V,A)$. Escreva uma função para imprimir o caminho entre dois vértices de um grafo baseada nos vetores *ant*, que guarda os antecessores e no vetor *dist*, que guarda as distâncias, ambos obtidos após a execução da função de busca em largura.

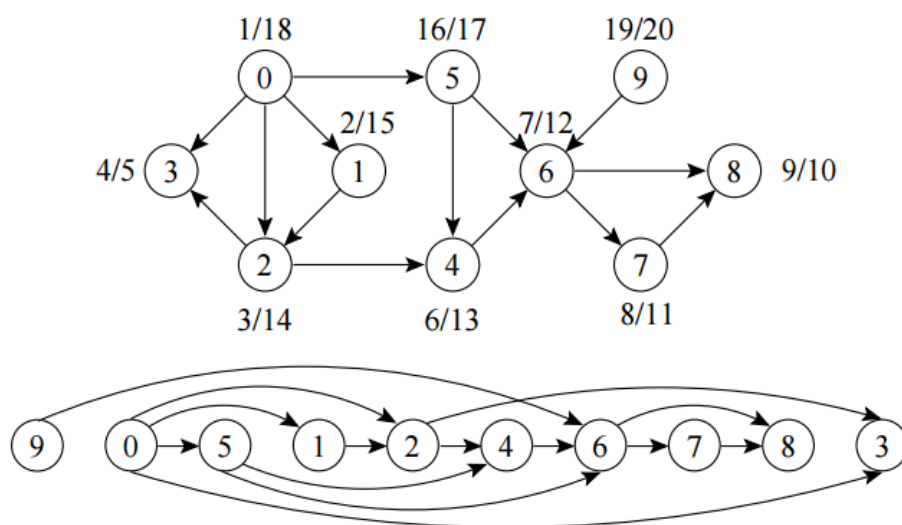


Figure 1: Exemplo de grafo com ordenação topológica. São exibidos os tempos.