

Algoritmos para Automação e Sistemas
Sexta Lista de Exercícios

Grafos

1) Mostre os valores de d e π que resultam da execução da busca em largura sobre o grafo orientado da figura a seguir, usando o nodo cinza como origem.

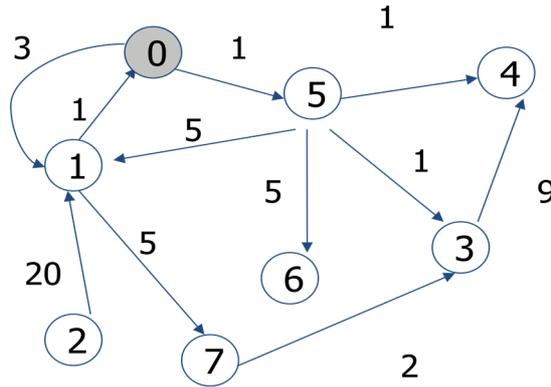


Figura 1: Grafo Orientado

2) Mostre os valores de d e π que resultam da execução da busca em profundidade sobre o grafo orientado da figura acima, usando o nodo cinza como origem.

3) Reescreva o procedimento DFS visto em sala de aula, utilizando agora uma pilha para eliminar a recursão.

4) Implemente um algoritmo que imprima como saída o resultado do caminhamento em largura e profundidade. Considere o seguinte grafo para validar a sua implementação:

- 0 1
- 0 2
- 1 3
- 1 4
- 2 3
- 3 4
- 3 5
- 5 6
- 5 7
- 6 7

Note que o seu programa deve ler um arquivo contendo pares de arestas. Por exemplo, 0 1 significa que existe uma aresta que conecta o vértice 0 ao vértice 1. Considere todas as seguintes formas de implementação para representar grafos:

- Listas de adjacências;
- Matriz de Adjacências;
- Listas de arestas.

Realize uma análise de espaço e tempo para cada forma de implementação.

5) Implemente um algoritmo que, dado um grafo dirigido, retorne todos os seus respectivos ciclos. Tal algoritmo deve receber como formato de entrada uma matriz binária representando o grafo. Por exemplo:

Entrada:

```

  A B C D E F G H
A 0 0 0 1 0 0 0 0
B 1 0 0 0 0 0 0 0
C 0 1 0 0 0 0 0 0
D 0 0 1 0 1 0 0 0
E 0 0 0 0 0 0 0 1
F 1 0 0 0 1 0 0 0
G 0 0 0 0 1 1 0 0
H 0 0 0 0 0 0 0 1 0

```

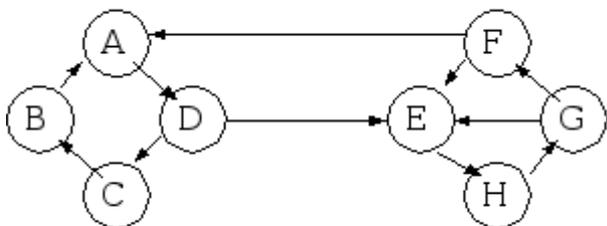
Saída:

A - D - C - B - A

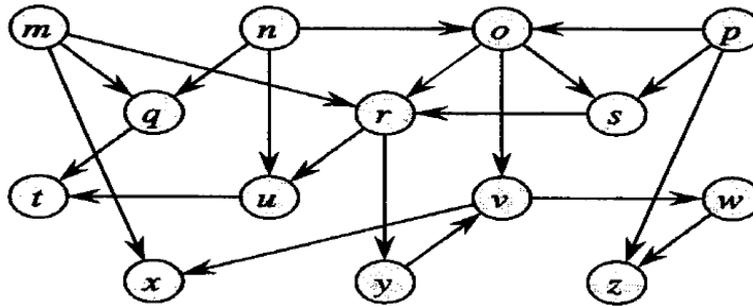
E - H - G - F - E

A - D - E - H - G - F - A

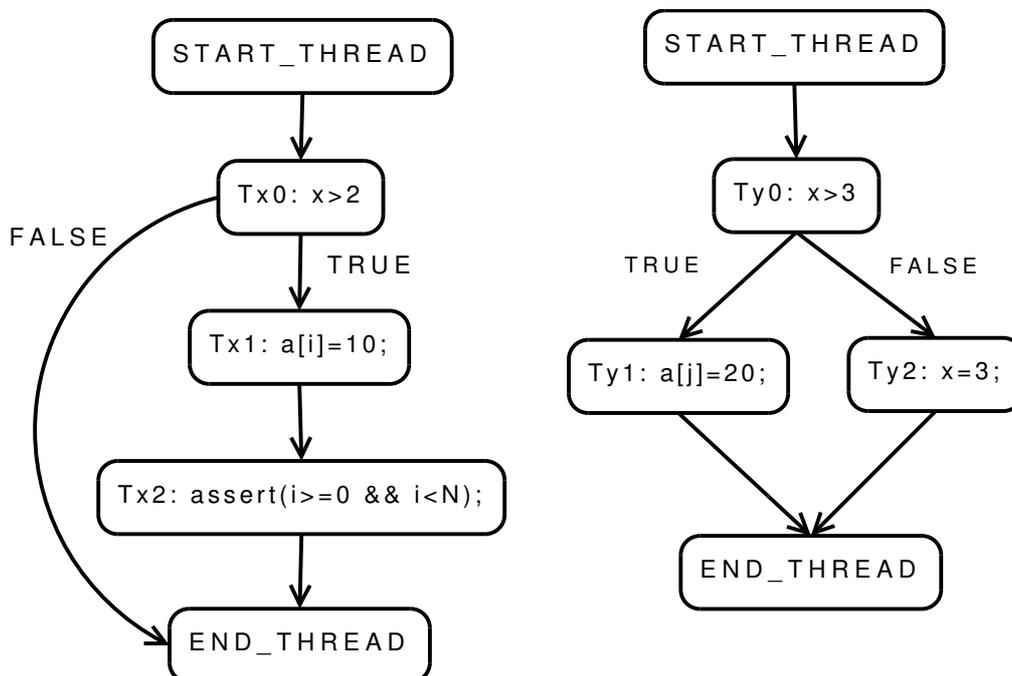
E - H - G - E



6) Implemente um algoritmo que mostre a ordenação de vértices produzido pelo procedimento TOPOLOGICAL-SORT, quando o mesmo é executado sobre o grafo a seguir:



7) Considere o seguinte programa multitarefa com dois processos, onde cada processo é modelado usando grafo de fluxo de controle (ou CFG, do inglês *Control-Flow Graph*).



Os comandos `START_THREAD` e `END_THREAD` iniciam e terminam a execução de uma tarefa, respectivamente. Cada nodo do grafo contém um rótulo para identificação e um comando para ser executado. Note que as variáveis globais “x” e “a” são compartilhadas entre as duas tarefas. Desta forma, este programa multitarefa pode produzir diferentes linhas de execução (i.e., intercalações) dependendo do algoritmo de escalonamento utilizado. O número de intercalações é exponencial no número de tarefas e comandos (conforme visto em sala de aula). De posse destas informações, você deve responder as seguintes questões:

- Implemente um grafo de fluxo de controle para representar cada tarefa.
- Construa uma árvore para representar todas as possíveis intercalações deste sistema a partir do CFG das tarefas.

8) Use a árvore produzida no exercício 7) para realizar o caminhar em largura com o propósito de imprimir todas as possíveis intercalações.

9) Use árvore produzida no exercício 7) para realizar o caminhamento em profundidade com o intuito de imprimir todas as possíveis intercalações.

10) Implemente um algoritmo usando tabelas hash para eliminar as intercalações redundantes que resultam da execução do programa multitarefa do exercício 7). Qual seria a complexidade da sua solução?

Data de entrega: 21 de julho de 2015 (terça-feira).

Após esta data será descontado 2 pontos por dia de atraso.

A lista de exercícios deve ser resolvida e entregue individualmente.

14/07/2015