MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Alfenas . UNIFAL-MG
Rua Gabriel Monteiro da Silva, 714 . Alfenas/MG . CEP 37130-000
Fone: (35) 3299-1000 . Fax: (35) 3299-1063

Prova 01 de Teoria dos Grafos - Prof. Humberto César Brandão de Oliveira
Semestre: 2010/2 - Data: 17/09/2010

Aluno: ____ Matrícula: ________________________ Nota:___________

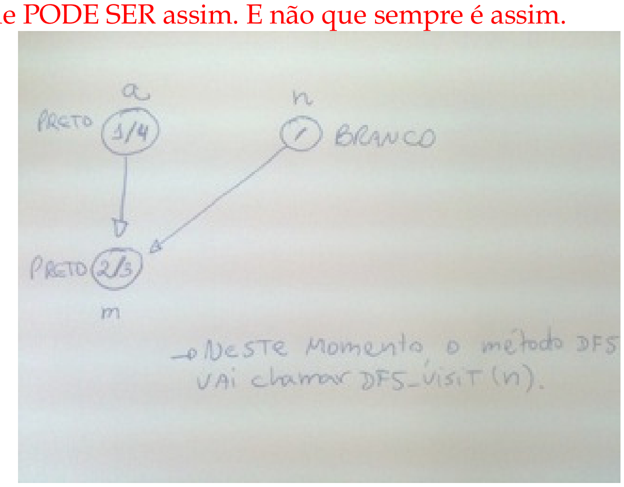
Questão	Valor	Nota
1	3	
2	3	
3	3	
4	3	
5	3	
6	3	
7	2	
Total	**20**	

1. Verdadeiro ou falso. Justifique sempre!
    a. (0.5 pontos) O método DFS-VISIT pode ser chamado |A| vezes durante a busca em profundidade em um grafo G=(V,A);

Verdadeiro: o métdo DFS_VISIT é chamado |V| vezes na DFS. Se |A| = |V|, então o método DFS_VISIT é chamado |A| vezes, fazendo a afirmação verdadeira.

    b. (0.5 pontos) Durante a execução do método DFS, um vértice *m*, atingível a partir de *n* no grafo G=(V,A), pode ser PRETO, enquanto *n* é BRANCO.

Verdadeiro: Prova da afirmação na figura a seguir. Neste caso, o exemplo PROVA, porque a afirmação diz que PODE SER assim. E não que sempre é assim.



    c. (0.5 pontos) O algoritmo BFS é implementado com o auxílio de uma pilha, podendo ser a pilha de execução.

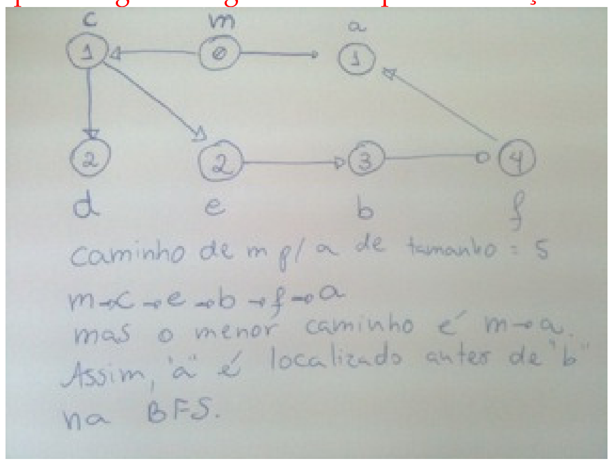
Falso: O algoritmo DFS é implementado com o auxílio de uma pilha, podendo ser a pilha de execução. O BFS é implementado utilizando uma fila.

    d. (0.5 pontos) A representação por lista de adjacência utiliza internamente uma matriz quadrada de grau |V|, para um grafo G=(V,A);

Falso: A matriz de adjacência é que utiliza uma matriz quadrada de grau |V|. A lista de adjacência utiliza um vetor de listas encadeadas para representação computacional do grafo.

    e. (0.5 pontos) Suponha o algoritmo BFS com o vértice *m* sendo o inicial para a busca. Suponha outros dois vértices: *a* e *b*. Se existe um caminho de *m* até *a* de tamanho 5, e um caminho de *m* até *b* de tamanho 3, então, *b* será localizado antes que *a* na BFS no grafo G=(V,A).

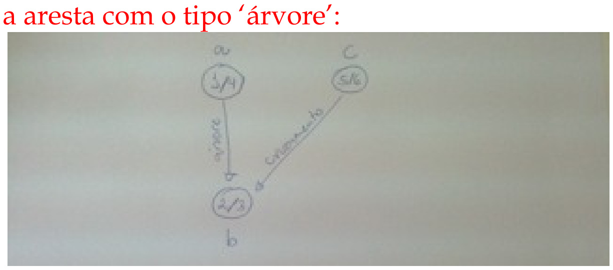
Falso: Existir um caminho de tamanho 5 de *m* até *a* não quer dizer que este é o menor caminho. Contra exemplo na figura a seguir mostra que a afirmação é falsa.



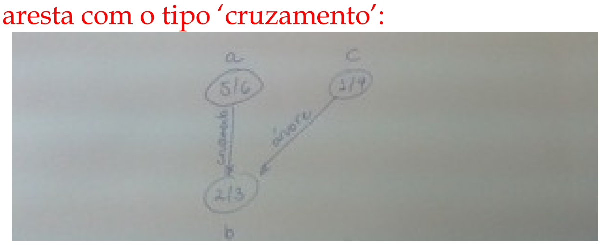
f. (0.5 pontos) Suponha o algoritmo DFS executando em um grafo G=(V,A) com o vértice *a* sendo o primeiro a ser marcado de cinza. Se a aresta *(a,b)* for marcada como aresta do tipo árvore, para qualquer outra DFS, começando de qualquer outro vértice, sempre a aresta *(a,b)* será do tipo árvore.

Falso: Contra exemplo a seguir, onde a aresta *(a,b)* atende aos requisitos da afirmação, e não é marcada como sendo do tipo árvore, quando a busca começa do vértice 'c'.

Exemplo marcando a aresta com o tipo 'árvore':



Exemplo marcando a aresta com o tipo 'cruzamento':



2. (3 pontos) Qual é a quantidade máxima de arestas de um grafo não direcionado que aceita no máximo 3 arestas paralelas para cada par de vértices. Considere *n* como o número de vértices do grafo G=(V,A).

$$\text{Arestas em um grafo simples completo} = \left(\frac{\sum_{i=1}^{n} (n\text{-}1)}{2} \right) = \frac{n^2 - n}{2}$$

$$\text{Grafo orientado com 3 arestas paralelas sem aresta laço} = 3\left(\frac{n^2 - n}{2}\right)$$

$$\text{Somente arestas laço} = 3n$$

$$\text{Grafo não orientado com 3 arestas paralelas} : 3\text{n} + 3\left(\frac{n^2 - n}{2}\right)$$

---

3. (3 pontos) Implemente o algoritmo de busca em profundidade iterativo (sem recursão).

Utilize a estrutura de dados pilha para simular a pilha de execução utilizada implicitamente na recursão.

4. (3 pontos) Implemente um algoritmo de busca em profundidade capaz de localizar ciclos em um grafo. Retorne verdadeiro se o grafo possui ciclo, e faço, e caso contrário;

Adaptação simples. Dentro do DFS_VISIT, antes de chamar o método recursivamente, verifique se o vértice em questão é cinza. Se for, retorne verdadeiro.
Se terminar a busca DFS, retorne falso.

$DFS(G)$
$\quad para\ cada\ vértice\ u \leftarrow V[G]$
$\qquad cor[u] \leftarrow BRANCO$
$\quad tempo \leftarrow 0$
$\quad para\ cada\ vértice\ u \in V[G]$
$\qquad se\ cor[u] = BRANCO$
$\qquad\quad DFS-VISIT(u)$
$\quad retorne\ \ falso$

$DFS-VISIT(u)$
$\quad cor[u] \leftarrow CINZA$
$\quad tempo \leftarrow tempo + 1$
$\quad d[u] \leftarrow tempo$
$\quad para\ cada\ vértice\ v \in Adj(u)$
$\qquad se\ cor[v] = CINZA$
$\qquad\quad retorne\ verdadeiro$
$\qquad se\ cor[v] = BRANCO$
$\qquad\quad DFS-VISIT(v)$
$\quad cor[u] \leftarrow PRETO$
$\quad f[u] \leftarrow tempo \leftarrow (tempo + 1)$

5. (3 pontos) Implemente um método capaz de imprimir o caminho do vértice de partida s, até um vértice u qualquer, após terminada a busca em largura. É permitido utilizar estruturas de dados produzidas pela busca em largura.

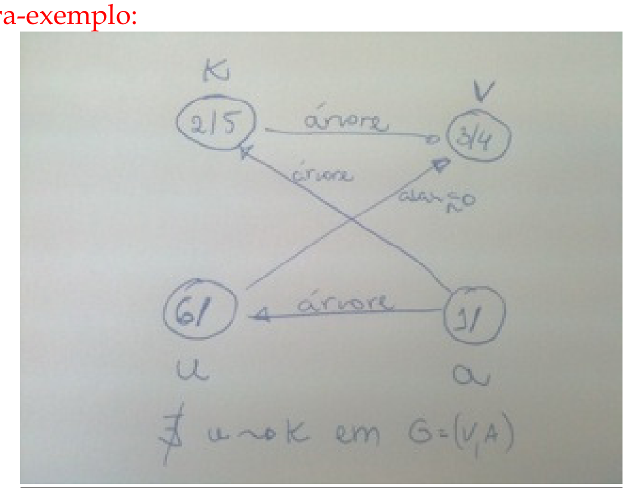
Dado em sala de aula.

MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Alfenas . UNIFAL-MG
Rua Gabriel Monteiro da Silva, 714 . Alfenas/MG . CEP 37130-000
Fone: (35) 3299-1000 . Fax: (35) 3299-1063

$$
\begin{array}{l}
PRINT\_PATH(G, s, v) \\
\quad if\ \ v = s \\
\qquad cout << s \\
\quad else\ if\ \ \pi[v] = NULL \\
\qquad cout << "não\ existe\ trajeto" \\
\quad else \\
\qquad PRINT\_PATH(G, s, \pi[v]) \\
\qquad cout << v \\
\quad end\ if \\
end\ \ function
\end{array}
$$

6. (3 pontos) Se existe uma aresta de avanço de *(u,v)* localizada enquanto *u* é cinza na busca em profundidade, e o vértice *k* está na mesma árvore do vértice *v* na floresta primeiro em profundidade, então existe caminho de *u* para *k* no grafo original. Verdadeiro ou falso? Prove.

Falso: Considere o contra-exemplo:



7. (2 pontos) Cite 3 aplicações que podem ser feitas com o algoritmo de busca em profundidade, ou de adaptações simples no algoritmo de busca em profundidade.
   - Busca de arquivos dentro de um sistema operacional;
   - Detecção de *deadlocks;*
   - Enumeração de todas as possibilidades para resolver problemas combinatórios.