Universidade Estadual de Maringá – Departamento de Informática
1747 – Teoria dos Grafos e Análise de Algoritmos
Ciência da Computação
Profs. Marco Aurélio e Mauro Henrique Mulati

Simulado 2

1. (1,0) Um *grafo bipartido* é um grafo não orientado $G = (V, E)$ em que V pode ser particionado em dois conjuntos V_1 e V_2 tal que, se $(u, v) \in E$ então $u \in V_1$ e $v \in V_2$ ou $v \in V_1$ e $u \in V_2$. Prove ou conteste: o complemento de um grafo bipartido é bipartido.

2. (1,0) A representação por *conjuntos* de um grafo utiliza duas listas: uma lista de vértices e uma lista de pares de vértices para representar as arestas. Para cada caso a seguir, explique como o algoritmo de busca em largura pode ser adaptado para funcionar com grafos de entrada representados por conjuntos e analise o tempo de execução do algoritmo adaptado.
   (a) A quantidade de memória utilizada pelo algoritmo, além da utilizada na representação dos conjuntos do grafo de entrada, deve ser constante.
   (b) Não existe restrição de memória.

3. (1,0) Dado uma árvore $T = (V, E)$ (representada por uma lista de adjacências) e o nó raiz $r \in V$, crie um algoritmo para pré-processar a árvore de maneira que perguntas da forma "u é um ancestral de v?" possam ser respondidas em tempo constante. O pré-processamento deve tomar tempo linear. Lembre-se de que u é considerado um *ancestral* de v na árvore T com raiz r, se o caminho de r até v em T passar por u.

4. (3,0) Escreva um algoritmo com tempo de execução $O(V \cdot E)$ que determine se um grafo não orientado $G = (V, E)$ contém um ciclo de tamanho 3. Explique como seu algoritmo funciona e faça a análise do tempo de execução.

5. (4,0) Uma *articulação* de um grafo não orientado é um vértice cuja a remoção aumenta o número de componentes do grafo. Escreva um algoritmo de tempo linear que identifique todas as articulações de um grafo. Explique como o seu algoritmo funciona e faça a análise do tempo de execução.