1 Prova de Grafos

1. Um grafo não dirigido conectado é dito *bipartido* se todos os seus vértices podem ser particionados em dois subconjuntos disjuntos X e Y; de modo que toda aresta conecta um vértice em X a um vértice em Y. Projete um algoritmo baseado em pesquisa primeiro em largura para verificar se um grafo é bipartido ou não.

2. Mostre como o algoritmo de Kosaraju-Sharir funciona sobre o grafo mostrado na Figura 1 (seja detalhista). Considere os vértices em ordem alfabética invertida e que as listas de adjacências também estejam em ordem alfabética invertida.

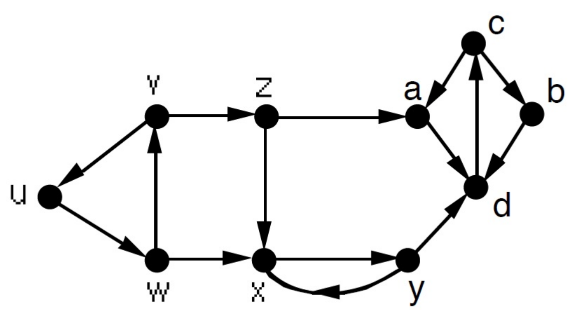

Figura 1:

3. A Ufes e várias universidades do Brasil estão participando um projeto conjunto. A coordenação do projeto resolveu que uma rede de computadores de altíssimo desempenho será construída conectando as entidades participantes. Os elos dessa rede formaram uma árvore. Também foi decidido que um servidor de aquivos será instalado em uma das universidades. Uma vez que o tempo de transmissão em um elo é dominado pelo tempo de sincronização, nós temos que o tempo de transferência de dados é proporcional ao número de elos utilizados. Desta forma é desejável escolher uma localização central para o servidor de arquivos. Dado uma árvore T, a excentricidade de um nó v de T é o caminho mais longo de v para qualquer outro nó de T. Um nó com excentricidade mínima é chamado de centro de T.

   Projete um algoritmo eficiente, em pseudocódigo, o qual toma como entrada uma árvore T com n nós, determina um centro de T. (Explique como o seu algoritmo funciona.)

4. Considere o seguinte problema:

   Problema K CONJUNTOS NO PLANO
   Entrada: Um conjunto de N pontos distintos no plano, a distância entre os pontos é a distância Euclidiana..
   Saída: K conjuntos de pontos $A_1, \ldots, A_k$, *tal que* $A_1 \cup A_2 \cup \ldots \cup A_k = N$, $A_i \cap A_j = \emptyset$, *para todo* $i \neq j$, *e cada ponto de qualquer conjunto* A_i *está tão perto ou mais perto de algum outro ponto de* A_i *que de qualquer outro ponto de* $N \setminus A_i$.

   (a) Projete um algoritmo eficiente, em pseudocódigo, que resolva o problema K CONJUNTOS NO PLANO. (Explique como o seu algoritmo funciona.)

   (b) Implemente o seu algoritmo em Sage. A entrada será o valor de K e uma lista de pontos L, e a saída será os pontos particionados em K conjuntos. Por exemplo,

   K = 2
   L = {(0, 0),(3, 2),(1, 2.5),(2, 2),(5, 1),(1, 1)}

A1 = {(0, 0),(3, 2),(1, 2.5),(2, 2),(1, 1)}
A2 = {(5, 1)}