

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**Faculdade da Computação**

**Trabalho de AED2 – Valor 15 Pontos**

**Prof. Luiz Gustavo Almeida Martins**

- Pode ser realizado em grupo de **no máximo 3 alunos**.
- Deve ser entregue em mídia (ex: DVD) **ATÉ o dia 28/06/2017**.
- Os códigos deverão ser implementados em Linguagem C e utilizar os TADs implementados pelos alunos (está vetado o uso de bibliotecas de grafos prontas).

### Problema do Roteamento Multicast

Uma transmissão de rede pode ser do tipo unicast, multicast ou broadcast. Em transmissões unicast conecta-se um ponto da rede a um outro ponto qualquer, para fazer isso de forma eficiente basta encontrar o melhor caminho entre os dois pontos. As comunicações broadcast caracterizam-se pelo fato de um nó da rede (servidor) enviar o conteúdo a todos os demais, para obter as melhores rotas para trafegar os dados, basta verificar a árvore geradora de custo mínimo. As transmissões multicast desejam, a partir de um nó da rede, transmitir o conteúdo para alguns outros, o que apresenta maior complexidade, pois é necessário obter uma árvore de Steiner de custo mínimo, o que é mais difícil que calcular uma única rota ou construir a árvore geradora de custo mínimo.

O problema do roteamento multicast (PRM) consiste em construir uma árvore de Steiner de custo mínimo a partir do grafo que representa a rede, a árvore deve ter raiz no nó correspondente ao servidor (nó raiz da árvore) e incluir todos os vértices destinos da transmissão multicast (nós folha da árvore). Não é trivial construir tal árvore, na verdade, este é um problema np-completo [15] e obter a solução exata se torna inviável a medida que a complexidade da rede aumenta.

Dado um grafo  $G=(V,E)$ , um nó raiz  $s \in V$  (nó transmissor) e um conjunto de nós destinos  $D \subset V$  (nós receptores), o PRM consiste em determinar um caminho de  $s$  que inclui todos os vértices em  $D$  e apresenta o menor custo possível.

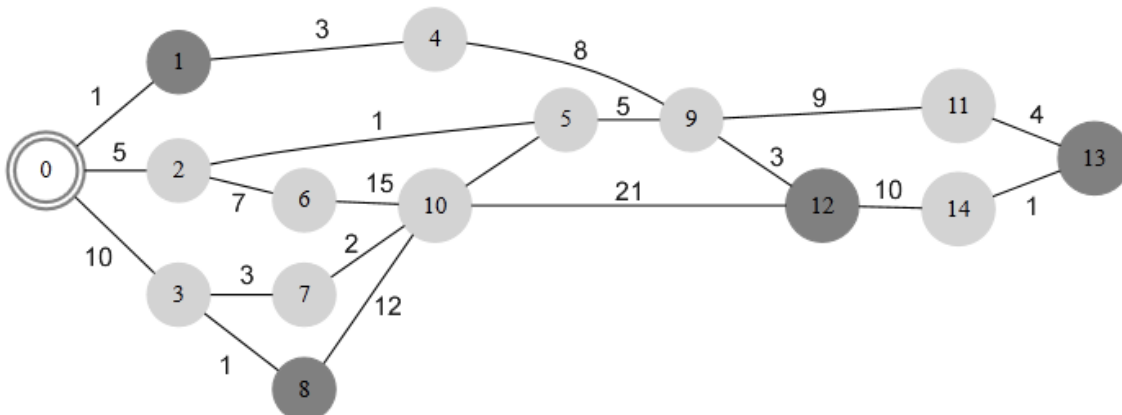


Figura 1: Exemplo de grafo G.

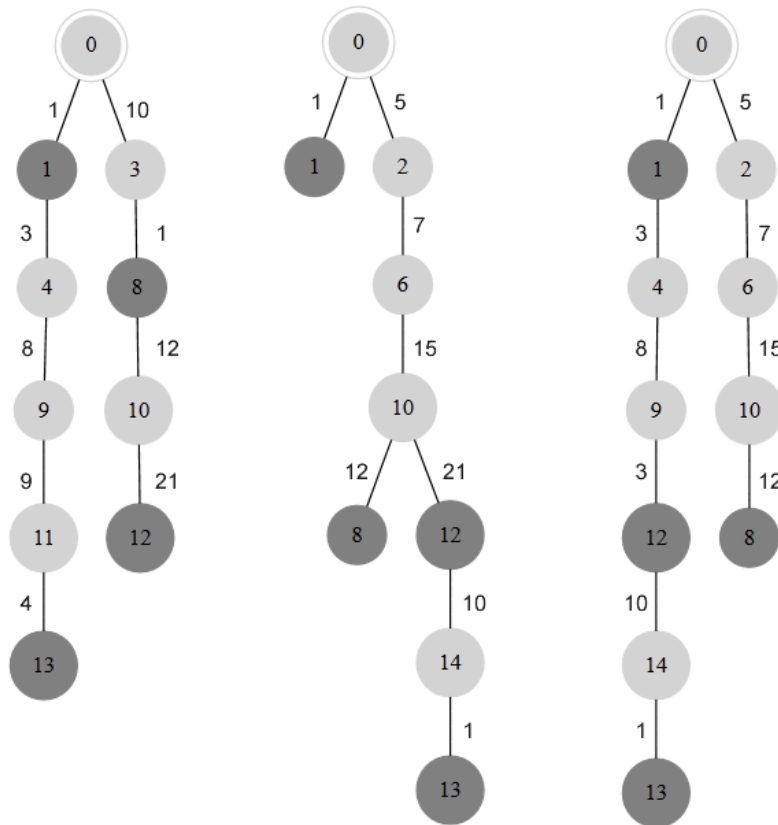


Figura 2: Exemplos de árvores multicast relativos ao grafo da figura 1.

Neste trabalho serão considerados quatro valores de peso para as arestas:  $c(e)$ ,  $d(e)$ ,  $z(e)$  e  $t(e)$ , os quais correspondem ao custo, delay, capacidade e tráfego atual do link, respectivamente. A partir desses valores, é possível computar as seguintes métricas:

- **Custo total da árvore:** soma dos custos de todas as arestas da árvore multicast

$$\text{custo}(T) = \sum_{e \in E_t} c(e)$$

- **Delay fim-a-fim máximo:** soma dos delays das arestas do caminho da árvore multicast com maior delay

$$D_{\max}(T) = \max_{v \in D} \text{Delay}(v)$$

- **Utilização máxima do enlace:** maior valor da razão entre o tráfego atual (acrescido do tamanho da mensagem  $\Phi$ ) e a capacidade de cada enlace.

$$\alpha_{\max}(T) = \max_{e \in E_t} \frac{t(e) + \phi}{z(e)}$$

## Parte 1 – Modelagem do grafo

Modele um grafo para representar uma rede para o PRM.

- Cada ponto da rede deve ser representado por um nó.
- As arestas indicam que há uma rota entre dois pontos e devem guardar os valores dos 4 pesos.
- Crie um arquivo texto contendo a representação textual do grafo da Figura 1.

## Parte 2 – Implementação do TAD grafo

Implementar o TAD Grafo não direcionado, de acordo com as especificações abaixo:

- Utilizar nas 2 formas de implementação (matriz de adjacências e listas de adjacências).
- Incluir a função **int numVertices(grafo \*g)**, a qual retorna o número de vértices que compõem o grafo.
- Incluir a função **int grauVertice(grafo \*g, int v)**, a qual retorna o grau do vértice especificado.
- Incluir a função **int ehAdjacente(grafo \*g, int v1, int v2)**, a qual retorna 1 se v1 for adjacente a v2, e 0 caso contrário.
- Implementar uma função para ler esse arquivo texto e contruir o grafo correspondente.
- Implementar outra função capaz de gerar o arquivo texto a partir de um grafo existente.
- Implementar um algoritmo baseado no Dijkstra para o cálculo do melhor caminho para enviar uma mensagem entre o ponto emissor para cada um dos pontos receptores, de acordo com uma das 4 métricas. Além da métrica desejada, na chamada da função deve ser informado o ponto de origem, os pontos de destino e o tamanho da mensagem. O algoritmo deve mostrar todos os pontos que a mensagem passará (árvore multicast) e o valor resultante das 4 métricas nesta transmissão.
- Repetir a implementação anterior, considerando o percurso em profundidade e em largura.
- Implementar um programa de usuário que forneça um menu para acesso às funções descritas acima.
- A partir dos resultados obtidos com cada uma dos métodos de percorrimento, analisar o comportamento das métricas e discutir sobre o desempenho dos métodos de acordo com cada métrica individualmente e de forma geral (considerando todas as métricas). Neste contexto, verifique se mudanças na configuração da transmissão multicast (diferentes pontos emissores, pontos receptores e tamanhos da mensagem) afetam o comportamento observado. Essa análise deve estar em um **arquivo chamado analise\_metodos.pdf**.
- Compare o desempenho dos métodos de percorrimento de acordo com cada forma de implementação do grafo. Teste as mesmas configurações (pontos emissores, pontos receptores e tamanhos da mensagem) usadas na análise anterior. Apresente o resultado da análise em um **arquivo chamado analise\_implementacao.pdf**.