UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade da Computação

Trabalho de AED2 – Valor 10 Pontos

Prof. Luiz Gustavo Almeida Martins

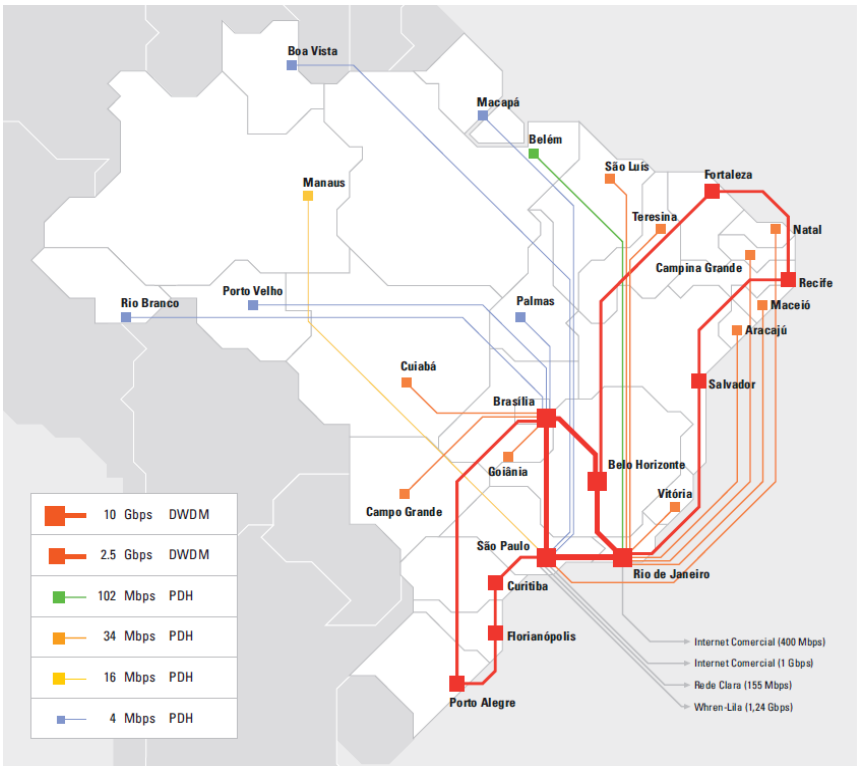
- Deve ser realizado em **grupo de 3 alunos**.
- Deve ser entregue em mídia (ex: CD ou DVD) ATÉ o dia **15/05/2018**.
- Junto com o trabalho devem ser entregues as implementações das listas de exercícios individuais realizadas até o momento (cada aluno deve ter sua pasta), devidamente separadas em subdiretórios (ex: /Joazinho/Lista1_AnaliseAlgoritmos/, /Zezinho/Lista2_Recursividade/, etc.).
- Os códigos deverão ser implementados em Linguagem C e utilizar os TADs implementados pelos alunos (está vetado o uso de bibliotecas de estruturas prontas).

Parte 1 – Modelagem utilizando grafos

Modelar a rede avançada Ipê da RNP.

“A rede Ipê é uma infraestrutura de rede Internet voltada para a comunidade brasileira de ensino e pesquisa. Nela conectam-se as principais universidades e institutos de pesquisa do país, beneficiando-se de um canal de comunicação rápido e com suporte a serviços e aplicações avançadas.”

(Fonte: https://memoria.rnp.br/ipe/)



- Cada Pontos de Presença (PoP) deve ser representado por um nó.
- As arestas conectam os nós e indicam que há uma rota entre dois PoP
- Defina, para cada nó, sua posição geográfica (coordenada x, y). Defina como coordenada zero o canto inferior esquerdo do mapa. Use a régua para definir a posição geográfica dos nós. Outra opção é utilizar as coordenadas geográficas (latitude e longitude) de cada cidade (essa informação pode ser obtida na web).
- De um nome para cada nó, correspondendo ao nome do PoP.
- Crie um arquivo texto contendo um número sequencial para o nó, o seu nome e sua respectiva posição geográfica (pontos x e y ou coordenadas geográficas).
- Crie um outro arquivo texto contendo as conexões entre os nós. Em cada linha, deve conter os números sequenciais dos dois nós que são conectados pela aresta. Na frente de cada conexão, coloque a capacidade da mesma (use uma única escala, por exemplo, Gibabits).
- Criar um programa para ler os arquivos texto e contruir o grafo correspondente.

Exemplo de criação do grafo

Formato do arquivo texto com os nós

Número do nó; Nome do nó; posição x; posição y

0; Manaus; 10; 43
1; Barcelos; 10; 45
2; Santarem; 30; 44
....

Formato do arquivo texto para arestas

Origem; destino; peso

0;1;10
0;4;100
0;3;0.250
3;4;5
....

Parte 2 – Implementação do TAD grafo

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

- Adequar o TAD implementado na lista de exercícios para o contexto deste trabalho. A forma de implementação adotada deve ser escolhida de acordo com as características da aplicação. Forneça um arquivo texto justificando sua escolha (Por que utilizou essa forma de implementação?).
- 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 o algoritmo de Dijkstra para o cálculo do caminho mais rápido para trafegar um arquivo entre PoPs da rede. O usuário deve informar o PoP de origem, o PoP destino e o tamanho do arquivo (em Mbytes). O programa deve mostrar todos os PoPs por onde passará o arquivo e o tempo total (em segundos) para a sua transmissão.
- Implementar um algoritmo que recebe o PoP de origem, o tamanho de um arquivo (em Mbytes) e o tempo máximo (em segundos) de transmissão. Esse algoritmo deve apresentar todos os pontos que receberão todo o arquivo sem exceder o tempo estabelecido.
- Implementar um algoritmo que, dado o PoP de origem e a quantidade máxima de pontos intermediários, apresente a cobertura de um envio, ou seja, o nome de todos os PoPs que receberiam o arquivo enviado.
- Implementar um programa de usuário que forneça um menu para acesso às funcionalidades descritas acima.