

1. Objetivo

Desenvolver programa em linguagem C que solucione o problema da **escolha da altura da carreta baú** para uso na movimentação de carga por uma empresa de logística utilizando modelagem por grafos.

2. Definição do problema

A ocorrência de entalamento de veículos de transporte de carga na passagem de túneis, pontes, viadutos, praças de pedágios etc. é um problema recorrente no Brasil devido ao legado das obras de engenharia, e que causa prejuízo à economia e compromete a segurança da população (Figuras 1 e 2).



Figura 1 - Caminhão entalado no túnel Ayrtton Senna em São Paulo



Figura 2 - Caminhão entalado no viaduto General Olímpio da Silveira, em São Paulo

Uma empresa de logística de carga leve que opera em todo o território nacional definiu cinco alturas padrões de **2.5, 3.0, 3.5, 4.0 e 4.5** metros, para as carretas baú de sua frota (Figura 3).



Figura 3 - Carreta baú definida pela empresa de logística e altura padrões

Estas carretas são utilizadas para movimentação de carga entre as centenas de centros de distribuição espalhados pelo Brasil. O software de logística da empresa tem registrado o limite máximo de altura em metros da menor passagem existente

nas rotas entre os centros de distribuição da empresa, como no exemplo da Figura 4. No caso de não existir uma passagem limitante de altura, a empresa cadastrou a altura do [gabarito vertical mínimo para vias expressas](#) estabelecido pelo [DNIT](#) que é **5.5 metros**. A área de TI desta empresa quer implementar uma nova função no sistema na qual serão informados dois centros de distribuição da empresa e o sistema responda qual é a maior carreta baú que possa ser utilizada sem causar atolamento mas também transportando o maior volume de carga possível entre os centros de distribuição.

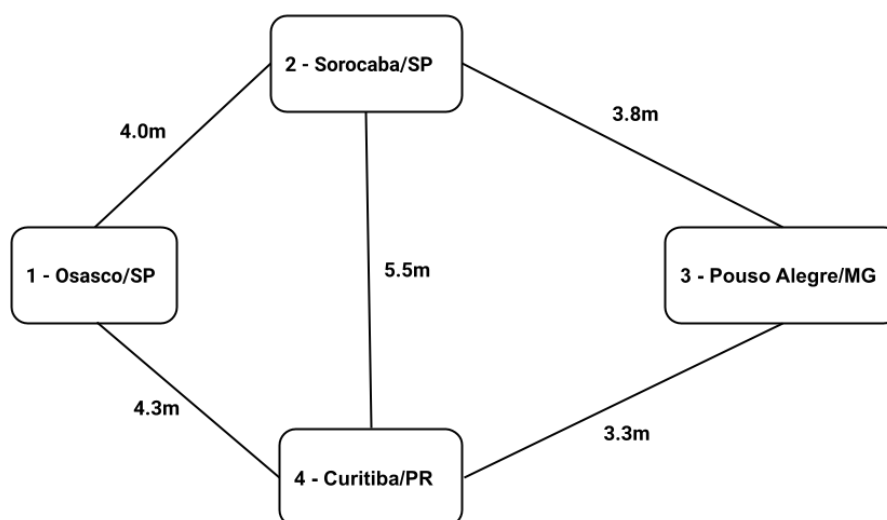


Figura 4 - Esquema exemplo com quatro centros de distribuição e limites em metros da menor passagem na rota entre centros de distribuição

3. Arquivos de entrada e saída

O arquivo de entrada contém várias linhas. Na primeira linha são informados o número de centros de distribuição da empresa, o número de rotas entre estes centros de distribuição e a quantidade de consultas desejadas.

Exemplo de entrada	Exemplo de saída
4 5 3 1 2 4.0 1 4 4.3 2 3 3.8 2 4 5.5 3 4 3.3 1 3	3.5 4.0 4.0

1 4	
1 2	

O exemplo acima é baseado na Figura 4. Na primeira linha: 4 centros de distribuição, 5 rotas entre estes centros, e 3 consultas são desejadas. As cinco linhas seguintes informam os limites máximos de altura existentes nas rotas entre dois centros de distribuição. Por exemplo, a linha 2 indica que na rota entre o centro 1 (Osasco/SP) e o centro 2 (Sorocaba/SP) existe um limite de altura de passagem de 4 metros. Todos as rotas diretas definidas entre dois centros de distribuição devem ser informadas, por exemplo, existe uma rota direta definida entre o centro 2 (Sorocaba/SP) e o centro 4 (Curitiba/PR), mas não existe uma rota direta definida entre o centro 1 (Osasco/SP) e o centro 3 (Pouso Alegre/MG). As rotas são definidas segundo critérios e fatores logísticos da empresa. O limite de altura existente entre um centro (por exemplo 2 - Sorocaba/SP) e outro (por exemplo centro 3 - Pouso Alegre/MG) é o mesmo na ida e na volta, portanto adota-se no arquivo de entrada a ordem crescente do número do centro de distribuição, ou seja, na linha 4 uma entrada **3 2 3.8** é inválida.

4. Itens obrigatórios

- Uso de grafos não direcionados;
- Uso de Makefile;
- Implementações por matriz e por lista de adjacências. A aplicação é a mesma (única), apenas altera-se o #include e com diferentes alvos no Makefile;
- Obrigatório o uso de interfaces para manipulação de grafos (como as definidas em sala, não necessariamente exatamente igual); pode-se incluir outras funções, por exemplo, `obtemVerticeDestino`.

5. Definições para entrega

- O programa **deve obrigatoriamente** aceitar como [argumento](#) um nome de arquivo de entrada, por exemplo, chamado **entrada.txt**;
- O programa **deve obrigatoriamente** gerar um arquivo de saída chamado **saida.txt**, no mesmo diretório onde o programa é executado;
- **Não utilize redirecionamento** da entrada-padrão com o caractere "<" (menor) ou da saída-padrão com o caractere ">" (maior);
- **Os arquivos podem ser carregados ("uploaded") individualmente ou "zipados" no e-disciplinas**, sem pastas ou subpastas. **Obrigatoriamente**, utilize os nomes dos arquivos definidos em sala de aula (aulas 3, 4 e 5) para a manipulação dos grafos:
 - **grafo_matrizadj.c** e **grafo_matrizadj.h**;
 - **grafo_listaadj.c** e **grafo_listaadj.h**;
- O arquivo executável gerado (alvo) no **Makefile** deve ser nomeado **ep1_matriz_99999999.exe** ou **ep1_lista_99999999.exe**, onde **99999999** é o número USP do estudante:

- `$make ep1_matriz_99999999.exe`
- `$/ep1_matriz_99999999.exe entrada.txt`
- `$make ep1_lista_99999999.exe`
- `$/ep1_lista_99999999.exe entrada.txt`

- O trabalho é **INDIVIDUAL**, com o nome do aluno e número USP devidamente identificado nas primeiras linhas do código-fonte.
- A entrega será realizada em atividade específica para o EP1 no ambiente e-disciplinas, até **a data limite informada no cabeçalho deste documento;**
- Deverá ser entregue os arquivos com os programas (*.c) e *headers* (*.h) e o **Makefile** utilizado para gerar o executável;
- Trabalhos com evidências de plágio serão desconsiderados e os autores receberão nota **ZERO**.