

Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados I (INF09292)
1º Trabalho Prático
Período: 2013/1
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Data de Entrega: 26/07/2013

Grupos de 2 pessoas

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Lista.

Regras Importantes

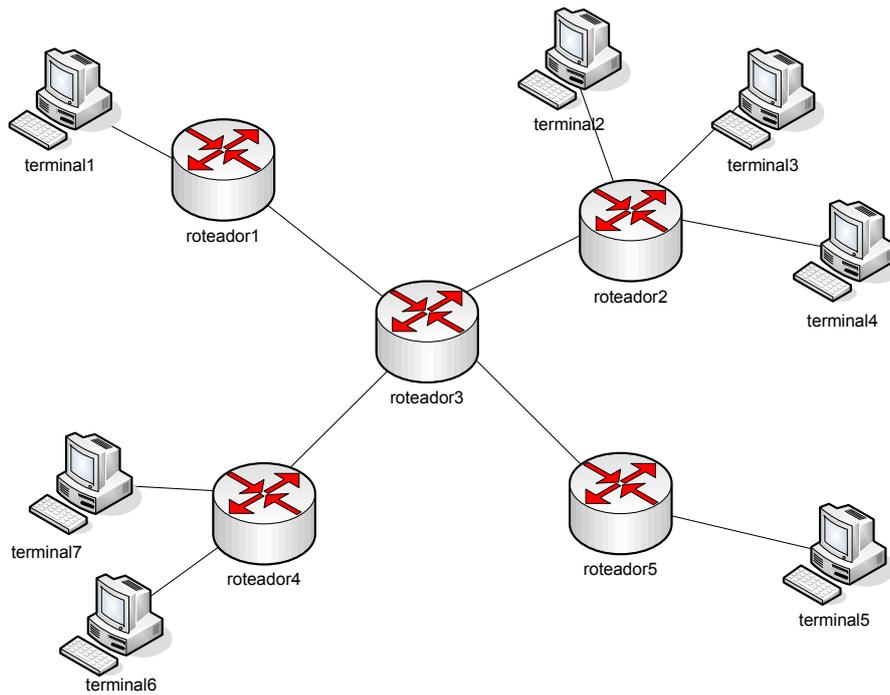
- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Para cada dia de atraso, é retirado um ponto da nota do trabalho.

Material a entregar

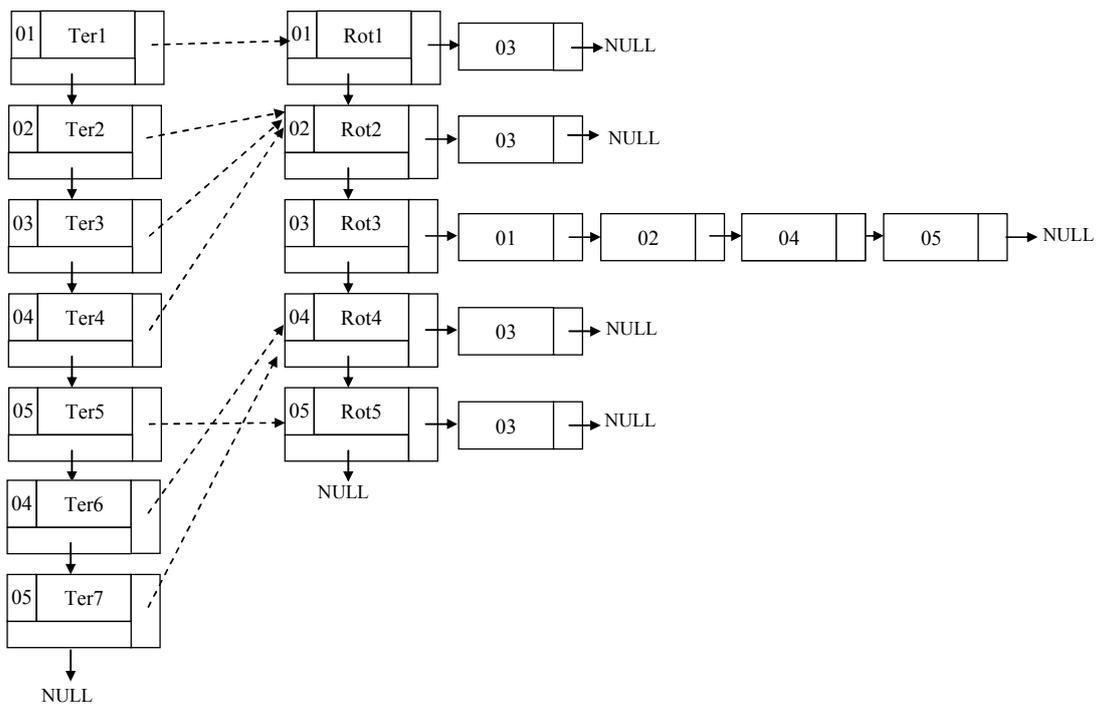
- Impresso: Documentação do trabalho, que deve conter:
 - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 - Implementação: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência a diagramas ilustrativos), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa usando Tipos Abstratos de Dados, como discutido em sala de aula (com tipos opacos).
 - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
- Por email (pdcosta@inf.ufes.br):
 - O assunto da mensagem deve ser ed20131EC:trab1:<nome1>:<nome2>
 - Por exemplo: ed20131EC:trab1:<joaosilva>:<mariacosta>
 - Documentação do trabalho (em formato PDF).
 - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
 - O makefile.
 - Favor nomear os arquivos da seguinte maneira: **roteadores.h**, **roteadores.c**, **enlaces.h**, **enlaces.c**, **terminais.h**, **terminais.c** e **NetMap.c**.

NetMap

NetMap é uma rede que permite a passagem de dados entre terminais distribuídos. Assim como a internet, NetMap é composta por roteadores que se conectam através de enlaces que habilitam a passagem de dados pela rede. Conectados a roteadores, estão os terminais. Cada terminal pode estar conectado a somente um roteador. Já um roteador pode estar conectado a vários terminais. A figura a seguir ilustra os componentes de um NetMap e a relação entre eles.



Como pode ser visto na figura, neste NetMap existem cinco roteadores e sete terminais. Os roteadores 1 e 3 estão conectados diretamente por um enlace, assim como os roteadores 2 e 3, roteadores 3 e 5 e roteadores 3 e 4. O NetMap pode ser implementado usando um conjunto de listas encadeadas: basicamente, implementa-se uma lista de Roteadores, onde cada célula contém os dados do roteador (identificador, nome e operadora). Além disso, cada célula desta lista de roteadores contém uma lista encadeada indicando quais são os roteadores conectados a ele, ou seja, representando os enlaces. Para implementar os terminais, pode-se implementar uma lista de terminais, na qual cada terminal possui um ponteiro para o seu roteador. Isso pode ser observado na seguinte figura:



Nesse trabalho, você deverá implementar essa estrutura. Considere que cada roteador é uma estrutura contendo os campos nome, identificador único e a operadora do roteador (e.x., telemar, gvt., etc.), e que não há roteadores ou terminais com os mesmos nomes. De maneira similar, considere também que os terminais são registros contendo os campos nome, identificador único e localização do terminal. O NetMap não aceita nomes compostos, tanto para nomes quanto para cidades. As seguintes operações devem ser implementadas¹:

- `CadastraRoteador(Roteador)`: cadastra um novo roteador ao NetMap. O identificador único desse roteador deve ser gerado automaticamente.
- `CadastraTerminal(Terminal)`: cadastra um novo terminal ao NetMap. O identificador único desse terminal deve ser gerado automaticamente.
- `RemoveRoteador(Roteador)`: remove o roteador do NetMap. Note que todas as relações deste roteador com outros devem ser excluídas.
- `ConectaTerminal(Terminal, Roteador)`: Conecta um terminal a um roteador, ambos já cadastrados no NetMap.
- `DesconectaTerminal(Terminal)`: Desconecta o terminal do seu respectivo roteador.
- `RemoveTerminal(Terminal)`: remove o terminal do NetMap.
- `ConectaRoteadores(Roteador1, Roteador2)`: Cria um enlace entre dois roteadores já cadastrados.
- `DesconectaRoteadores (Roteador1, Roteador2)`: Remove o enlace entre dois roteadores já cadastrados.
- `FrequenciaTerminal(localização)`: Imprime o número de terminais cadastrados de uma determinada localização.
- `FrequenciaOperadora(operadora)`: Imprime o número de roteadores cadastrados de uma determinada operadora.
- `EnviaInfo(Terminal1, Terminal2)`: Verifica se é possível enviar dados do Terminal1 para o Terminal2. Retorna SIM se for possível e NÃO caso contrário. Por exemplo, `EnviaInfo (Terminal1, Terminal5)` retorna SIM.
- `ImprimeDadosRoteador(Roteador)`: Imprime os dados de um roteador (nome, identificador, operadora), bem como o nome, o identificador e a cidade de todos os roteadores conectados a ele. Imprime também as informações dos terminais conectados a ele.
- `ImprimeDadosTerminal(Terminal)`: Imprime os dados de um terminal (nome, identificador, localização), bem como o nome, identificador e a operadora do roteador conectado a este terminal.
- `ImprimeTudo()`: imprime os dados de todos os roteadores cadastrados seguido da lista de roteadores conectados a ele (com nome e operadora de cada um), e a lista de terminais conectados (nome do terminal, localização).

A implementação da estrutura deverá ser feita utilizando alocação dinâmica de memória (ponteiros). Você deve fazer testes de consistência em **todas** as operações (teste de pré-condições) e deve imprimir mensagens de erro (por exemplo, a chamada de função `RemoveRoteador(Rot7)` no exemplo acima deve gerar uma mensagem de erro pois não há Roteador7 cadastrado com esse nome). Nas operações de inserção em listas encadeadas, insira o novo elemento ao final da lista. Além disso, procure escrever funções auxiliares que facilitem a implementação das operações acima, evitando a repetição desnecessária de código. Por exemplo, funções de manipulação de listas, pesquisa de id por nome, etc.

¹ As operações listadas aqui não representam as assinaturas das funções da implementação. Essas operações apenas indicam as principais funcionalidades que devem ser implementadas pelo seu programa, sem detalhes da implementação. Portanto, os retornos e os parâmetros das funções devem ser definidos pelo seu programa.

Organize o seu sistema usando o conceito de **Tipos Abstratos de Dados** discutido em sala de aula. Implemente 3 TAD's: roteadores.h, terminais.h e enlaces.h. Projete cuidadosamente as operações de cada TAD, respeitando os conceitos de encapsulamento e abstração, como discutimos em sala de aula.

Entrada e Saída

O seu programa (NetMap) deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (Faz parte do trabalho descobrir como manipular arquivos e strings em C). Exemplo de execução do programa a partir da linha de comando:

```
NetMap entrada.txt
```

O arquivo de entrada é basicamente uma lista de comandos (um por linha) em formato texto. O último comando é a palavra FIM, que indica o final do arquivo. O formato a ser usado é exemplificado abaixo:

Exemplo de arquivo de entrada (entrada.txt)

```
CADASTRAROTEADOR Rot1 Telemar
CADASTRAROTEADOR Rot2 Gvt
CADASTRAROTEADOR Rot3 Claro
CADASTRAROTEADOR Rot4 Telemar
CADASTRAROTEADOR Rot5 Gvt
CADASTRAROTEADOR Rot6 Gvt
CADASTRATERMINAL Ter1 Vitoria
CADASTRATERMINAL Ter2 VilaVelha
CADASTRATERMINAL Ter3 Vitoria
CADASTRATERMINAL Ter4 Colatina
CADASTRATERMINAL Ter5 Mimoso
CADASTRATERMINAL Ter6 Vitoria
CADASTRATERMINAL Ter7 Serra
CADASTRATERMINAL Ter8 Serra
REMOVEROTEADOR Rot10
REMOVEROTEADOR Rot6
REMOVETERMINAL Ter8
CONECTAROTEADORES Rot1 Rot3
CONECTAROTEADORES Rot2 Rot3
CONECTAROTEADORES Rot4 Rot3
CONECTAROTEADORES Rot5 Rot3
CONECTAROTEADORES Rot2 Rot5
CONECTATERMINAL Ter1 Rot1
CONECTATERMINAL Ter7 Rot4
CONECTATERMINAL Ter6 Rot4
CONECTATERMINAL Ter2 Rot2
CONECTATERMINAL Ter3 Rot2
CONECTATERMINAL Ter4 Rot2
CONECTATERMINAL Ter5 Rot2
CONECTATERMINAL Ter9 Rot1
DESCONECTAROTEADORES Rot2 Rot5
DESCONECTATERMINAL Ter5
```

```
CONECTATERMINAL Ter5 Rot5
FREQUENCIAOPERADORA Gvt
FREQUENCIATERMINAL Vitoria
IMPRIMEDADOSROTEADOR Rot3
IMPRIMEDADOSROTEADOR Rot2
ENVIAINFO Ter1 Ter4
IMPRIMETERMINAL Ter7
IMPRIMETUDO
FIM
```

Os comandos `FREQUENCIAOPERADORA`, `FREQUENCIATERMINAL`, `ENVIAINFO`, `IMPRIMEDADOSROTEADOR`, `IMPRIMETERMINAL` e `IMPRIMETUDO` deverão imprimir as informações em um arquivos de saída, a ser criado pelo seu programa. O nome do arquivo de saída deverá ser *saida.txt*. Todas as mensagens de erro também deverão ser impressas no arquivo *saida.txt*.

Os comandos `FREQUENCIAOPERADORA` e `FREQUENCIATERMINAL` imprimem em linhas do arquivo de saída o número de roteadores de uma determinada operadora e o número de terminais de uma determinada localização, respectivamente. O comando `ENVIAINFO` imprime em uma linha SIM se for houver a possibilidade de enviar dados de um determinado terminal para outro e NAO caso contrário.

O comando `IMPRIMEDADOSROTEADOR`(Roteador) deverá imprimir em uma linha as informações daquele roteador (nome, identificador e operadora), separados por espaço. Todos os enlaces deste roteador serão impressos nas linhas subsequentes (um por linha), cada linha contendo o nome, o identificador e a operadora do roteador. Antes das informações dos enlaces, inclua uma linha com a string “Enlaces de” + nome do roteador. Seguindo, deverão ser impressos os terminais conectados a este roteador um por linha. Use a string “Terminais de” + nome do roteador. Depois de impressa as informações dos roteadores, imprima uma linha em branco. Considerando o exemplo acima, a sequência de comandos `IMPRIMEDADOSROTEADOR`(Rot3), `IMPRIMEDADOSROTEADOR`(Rot2), imprime as seguintes linhas no arquivo *saida.txt*:

```
Rot3 03 Claro
Enlaces de Rot3
Rot2 02 Gvt
Rot1 01 Telemar
Rot4 04 Telemar
Rot5 05 Gvt
Terminais de Rot3
```

```
Rot2 02 Gvt
Enlaces de Rot2
Rot3 03 Claro
Terminais de Rot2
Ter2 02 VilaVelha
Ter3 03 Vitoria
Ter4 04 Colatina
```

O comando `IMPRIMETERMINAL` deverá imprimir as informações de um terminal (nome, identificador e o roteador). O comando `IMPRIMETERMINAL` (Ter7) deve imprimir:

```
Ter7 07 Serra
Rot4 04 Telemar
```

O comando IMPRIMETUDO deverá imprimir as informações de todos os roteadores do NetMap usando o mesmo formato de saída da função IMPRIMEROTEADOR(roteador).

Considerando o arquivo de entrada dado acima, e que são gerados pelo programa identificadores únicos para os roteadores e terminais, espera-se o seguinte no arquivo de saída:

Arquivo de saída para o arquivo entrada.txt

```
Erro: Roteador Rot10 inexistente no NetMap
```

```
Erro: Terminal Ter9 inexistente no NetMap
```

```
2
```

```
3
```

```
Rot3 03 Claro  
Enlaces de Rot3  
Rot2 02 Gvt  
Rot1 01 Telemar  
Rot4 04 Telemar  
Rot5 05 Gvt
```

```
Rot2 02 Gvt  
Enlaces de Rot2  
Rot3 03 Claro  
Terminais de Rot2  
Ter2 02 VilaVelha  
Ter3 03 Vitoria  
Ter4 04 Colatina
```

```
SIM
```

```
Ter7 07 Serra  
Rot4 04 Telemar
```

```
Rot1 01 Telemar  
Enlaces de Rot1  
Rot3 03 Claro  
Terminais de Rot1  
Ter1 01 Vitoria
```

```
Rot2 02 Gvt  
Enlaces de Rot2  
Rot3 03 Claro  
Terminais de Rot2  
Ter2 02 VilaVelha  
Ter3 03 Vitoria  
Ter4 04 Colatina
```

```
Rot3 03 Claro
```

```
Enlaces de Rot3
Rot2 02 Gvt
Rot1 01 Telemar
Rot4 04 Telemar
Rot5 05 Gvt
```

```
Rot4 04 Telemar
Enlaces de Rot4
Rot3 03 Claro
Terminais de Rot4
Ter7 07 Serra
Ter6 06 Vitoria
```

```
Rot5 05 Gvt
Enlaces de Rot5
Rot3 03 Claro
Terminais de Rot5
Ter5 05 Mimoso
```

Ponto Extra: a equipe que implementar a funcionalidade que permite encontrar o **menor caminho** entre dois terminais receberá um ponto extra. Neste caso, a documentação do trabalho deve indicar que a funcionalidade foi implementada para que sejam incluídos testes na bateria de testes: MENORCAMINHO Ter1 Ter2, que deve gerar a seguinte saída para o exemplo acima: Ter1 Rot1 Rot3 Rot2 Ter2

BOM TRABALHO!