

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METRÓPOLE DIGITAL

Linguagem de Programação II e Estruturas de Dados II- 2022.2

Especificação do Trabalho Final

1. Introdução

O bairro de Parnatal fica na cidade Parnatalândia. Neste bairro, moram grandes magnatas da Computação, que enriqueceram desenvolvendo programas eficientes para solucionar problemas NP-difíceis, os problemas mais complicados da computação. Por ter surgido recentemente, nenhuma empresa fornece conexão à internet para este bairro. Mas também pudera, os moradores só aceitam conexão direta por fibra óptica, porque eles precisam de uma internet muito veloz para o trabalho. Em Parnatalândia, nenhuma empresa fornece esse serviço, pois é muito caro. Vendo que há uma grande demanda, a empresa Net.já, que fica na cidade de Quequintura, decidiu que irá estender a sua rede para atender a todos os clientes de Parnatal. Para tanto, a Net.já fez um estudo preliminar de todas as possíveis maneiras de conectar os n moradores de Parnatal usando fibra óptica. Seu representante então discutiu as opções com os moradores:

- Em Parnatal, temos n residências. Eu posso fornecer um link exclusivo para cada morador, mas essa abordagem tem um custo extremamente elevado, pois Quequintura fica muito distante de Parnatalândia. (Os moradores prontamente recusaram essa abordagem.)
- Minha proposta é fazer um link único para atender a todo o bairro. Esse link único chegará em uma das casas a serem atendidas, denominada por **cX**. (Esse ponto foi aceito pelos moradores, mas todo mundo queria ter o privilégio de receber o link direto).
- Para atender aos outros moradores, uma solução óbvia seria conectar todos diretamente à **cX**. Mas esta solução, ainda muito centralizada é problemática: tem custo alto de conexão dadas as distâncias variadas e longas das casas a uma **cX** qualquer. Além disso uma quantidade grande de fibra ótica representaria um maior custo de manutenção para a vida inteira, que seria pago pelos moradores. Por fim, ainda toda a inclusão de novo link de fibra significaria manutenção sempre em **cX** que eventualmente esgotaria sua capacidade de distribuição. E ... (Ele ia continuar a argumentar, mas os moradores, que entendiam tudo sobre grafos, o interromperam e concordaram em também não adotar essa abordagem).
- A outra opção seria manter o link único para **cX** e construir uma rede em formato de árvore conectando todos os moradores com o menor custo possível e de maneira que uma casa somente estivesse ligada diretamente à, no máximo, d casas. Essa restrição é necessária para minimizar pontos vulneráveis na rede e balancear a carga da rede. Isso porque, depois que definirmos o desenho da rede e, por conseguinte, seu custo, podemos adicionar mais alguns links diretos, para outras residências, de maneira a reduzir a probabilidade de falhas na rede. Mas, nesse momento, temos dois problemas principais:

- o qual casa receberá o primeiro link direto? o dada uma matriz $n \times n$ contendo o custo de se criar uma conexão entre duas casas quaisquer em Parnatal, como definir a rede de distribuição entre os moradores de forma a ter o menor custo possível?
- Um morador teve a seguinte ideia: criar a rede de distribuição de maneira que cada residência esteja conectada diretamente a, no máximo, d casas é um problema NPdifícil. Nós somos especialistas nisso! Então, aquele morador que resolver de forma correta e mais rapidamente possível o problema de desenhar essa rede será premiado com o link de fibra diretamente da Net.já. O que vocês acham?

Os moradores de Parnatal nem responderam, correram para seus clusters para codificarem o algoritmo e encontrarem a melhor solução o mais rapidamente possível! A tarefa de vocês é, pois, dada uma matriz $n \times n$ contendo o custo de se criar uma conexão entre duas casas quaisquer em Parnatal e um valor d , encontrar essa estrutura de conexão de menor custo (em forma de árvore) entre as n residências de Parnatal, de maneira que nenhuma residência tenha mais que d conexões diretas com outras residências. Esses dados serão informados via arquivo. O formato geral do arquivo e um exemplo desse arquivo de entrada seguem abaixo:

```
* Formato:
<valor de n> <valor de d>
<custo c1---c2> <custo c1---c3> <custo c1---c4> ... <custo c1---cn>
<custo c2---c3> <custo c2---c4> ... <custo c2---cn>
<custo c3---c4> ... <custo c3---cn>
...
<custo c<n-1>-cn>

* Exemplo:
5 2
5 10 15 2
21 2 45
53 12
13
```

Notem que a conexão de uma casa $c1$ para uma casa $c2$ é a mesma que a conexão de $c2$ para $c1$, por isso, na entrada é fornecido apenas o valor para $\langle c1, c2 \rangle$ (e não para $\langle c2, c1 \rangle$ que seria o mesmo).

2. Objetivos Específicos

Conforme já mencionado, este é um problema NP-difícil. Isso significa que não existe prova de que um algoritmo com complexidade em tempo polinomial possa encontrar sempre a solução correta para este problema. Portanto, o algoritmo que vocês irão implementar deverá:

1. Ler arquivo de entrada formato de acordo com exemplo;
2. Testar todas as possíveis maneiras de construir uma árvore de conexões desejada, atendendo à restrição no número de conexões diretas em cada vértice (no máximo d);

- a Gerar todas as árvores (soluções) sem garantir uma ordem de custo. É preciso então verificar se cada solução atende a restrição de conexão.
 - b Gerar todas as árvores (soluções) de maneira ordenada por ordem de custo, de modo que a primeira solução seja a árvore geradora mínima do grafo, a segunda solução será a segunda árvore geradora mínima do grafo, e assim por diante. É preciso então verificar se cada solução atende a restrição de conexão.
3. Ao final da execução, guardar em um arquivo a melhor solução válida encontrada, por meio das conexões que devem ser criadas entre as casas e por meio do custo dessa solução. O custo da solução é dado pelo somatório dos custos das conexões criadas.

3. Informações gerais

O projeto consiste no desenvolvimento de uma solução em Java explorando conceitos de orientação a objetos e estruturas de dados. O paradigma de orientação a objetos tende a facilitar o entendimento e a organização do código. Dessa forma, o projeto deverá ser feito por um grupo de no máximo três (3) alunos, que irá apresentá-lo perante uma banca formada pelos professores das disciplinas. Cada grupo terá **20 minutos** para fazer sua apresentação. Nesta apresentação, os grupos deverão entregar o material pedido bem como demonstrar o funcionamento do projeto. Além disto, deverão estar aptos a responder questões sobre o desenvolvimento do projeto.

Importante: Não será dada uma única nota ao grupo. Cada componente do grupo receberá uma nota de acordo com seu desempenho durante esta apresentação.

4. Resultados esperados

Os grupos deverão submeter os seguintes itens:

Programa: o grupo deverá mostrar o programa funcionando satisfatoriamente. Alguns dos itens pedidos poderão não ter sido implementados. A nota levará em conta o esforço do grupo em entregar parte do projeto funcionando. Erros podem acontecer, mas não devem comprometer o funcionamento geral do programa.

Fonte(s): deverá(ão) ser entregue(s), via SIGAA, o(s) fonte(s) do programa para avaliação. Deverá vir junto com o(s) fonte(s) um arquivo README.TXT contendo instruções de como se pode obter o executável a partir deste(s) fonte(s). Observação Importante: Fontes que sejam bem estruturados e legíveis, ou seja, bem-organizados e separados por arquivos.

Relatório: um relatório que descreva como o seu programa foi construído. Esse documento deverá conter algumas informações sobre as fases de análise e projeto (Ex: diagrama de classes) demonstrando a aplicação dos conceitos de orientação a objetos na estrutura da solução desenvolvida. Este documento também deverá contemplar informações sobre as estruturas de dados não-triviais utilizadas, a saber: que informações são armazenadas nas estruturas utilizadas e uma justificativa para a escolha de cada estrutura em particular. Por fim, o

documento deverá conter a *análise assintótica* do tempo de execução do algoritmo implementado em função do número de casas.

Apresentação: sua apresentação deverá detalhar o projeto de seu sistema (diagrama de classes), assim como detalhar os algoritmos implementados pelo grupo. Inclua em sua apresentação quaisquer outras informações que o grupo achar necessário e que ajude a explicar o trabalho desenvolvido.

5. Modelagem e Implementação

Seu sistema deverá ser modelado de acordo com as melhores práticas de orientação a objetos, e ser implementado com a linguagem de programação Java. Lembre-se de agrupar classes com funcionalidades semelhantes em pacotes. Além disso, será **obrigatório** o uso de:

1. Encapsulamento
2. Modularização e padronização
 - Nomeação de classes e métodos segundo padrões;
 - Separação das classes em pacotes
 - Design de Classes (responsabilidade e coesão)
3. Herança
4. Polimorfismo
5. Classes abstratas
6. Interfaces
7. Tratamento de exceções
8. Estrutura de dados simples (qualquer coleção)
9. Estruturas de dados avançadas (heap, conjuntos disjuntos, árvores balanceadas etc.)

6. Entrega e Avaliação

Seu grupo deverá submeter, via SIGAA, um arquivo compactado contendo:

- Código fonte do sistema desenvolvido, incluindo um documento README.TXT contendo instruções de como se pode compilar o código fonte.
- Apresentação em PDF (slides).
- Relatório Técnico, contendo pelo menos as seguintes seções:
 - Introdução: contendo uma breve descrição do problema abordado e do que será apresentado em seu relato;
 - Descrição da abordagem de solução do problema: Descrevendo o projeto OO (diagramas de classes), e destacando as decisões de projetos tomada, bem como identificando padrões de projeto aplicados;
 - Descrição geral das estruturas de dados utilizadas com explicação dos algoritmos utilizados;

- Conclusão;
- Referências;

O relatório deverá ser feito seguindo o template da Sociedade Brasileira de Computação (SBC) que pode ser encontrado no seguinte endereço:

<http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-ecapitulos-de-livros/878-modelosparapublicaodeartigos>

Tabela de pontuação para correção de LPII:

Codificação	Classes Abstratas e Interface	0,5
	Tratamento de Exceções	0,5
	Herança e Polimorfismo	1,0
Apresentação	Relatório	2,0
	Apresentação	2,0
Objetivos	Ler arquivo de entrada	0,5
	Apresentar solução que teste todas as possíveis árvores	1,0
	Apresentar solução que teste todas as possíveis árvores em ordem de custo	2,0
	Armazenar a solução válida de menor custo encontrada	0,5
	Interface Gráfica (ponto extra)	1,0
	Total de pontos	11,0

Tabela de pontuação para correção de EDBII:

Codificação	Utilizar estruturas de dados vistas na disciplina	1,0
Apresentação	Relatório	2,0
	Apresentação	2,0
Objetivos	Apresentar solução que teste todas as possíveis árvores	2,0
	Apresentar solução que teste todas as possíveis árvores em ordem de custo	2,0
	Justificar a escolha das estruturas de dados utilizadas	1,0
	Total de pontos	10,0