



ESTRUTURA DE DADOS

Aula 1 – Introdução

Prof. Rodrigo Maciel



Conteúdo

- Programação Orientada a Objetos (POO ou OO)
- Pilhas e Filas
- Listas Encadeadas
- Recursão
- Ordenação
 - Bubble sort
 - Selection sort
 - Insertion sort
 - Shell sort
 - Merge sort
 - Quick sort

Conteúdo

- Árvores Binárias de Busca
- Grafos
 - Busca em largura e profundidade
 - **Busca gulosa e A***
 - **Algoritmo de Dijkstra**

Objetivo: entender a teoria/implementar e testar essas estruturas de dados



Estrutura de Dados

Disciplina que estuda as técnicas computacionais para a organização e manipulação eficiente de quaisquer quantidades de informações.

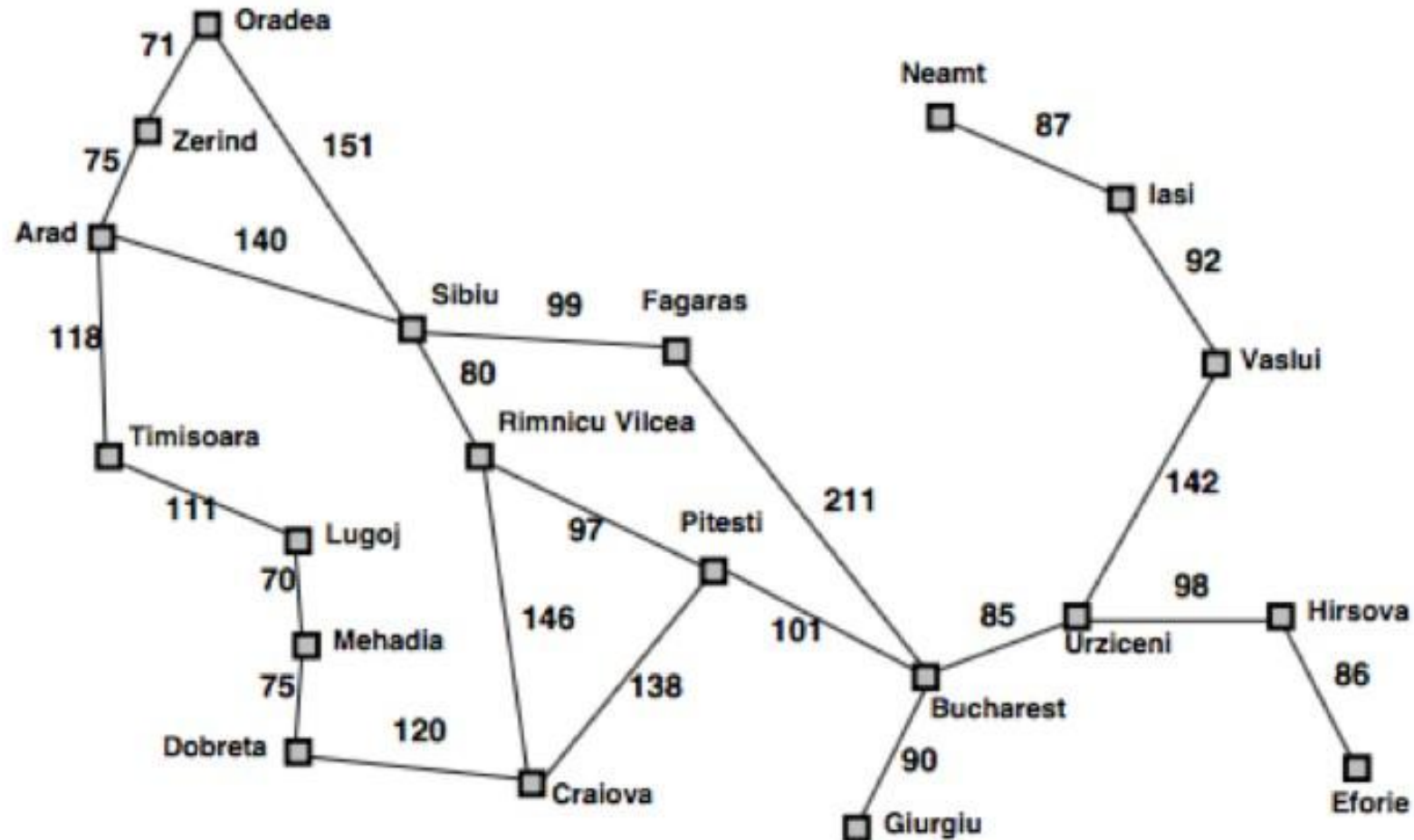
Estrutura de Dados - Problema 1

Considere uma rede de ferrovias conectando duas cidades através de um número de cidades intermediárias, onde cada ferrovia tem uma capacidade de transporte específica. Como encontrar o fluxo máximo entre as duas cidades?



Estrutura de Dados - Problema 2

Ir de Arad a Bucharest: encontrar o menor caminho





Estrutura de Dados

- Como representar estes problemas em computadores?
- Como construir os algoritmos necessários?
- Que estrutura de dados utilizar?
- De que forma os dados estão organizados?
- Que operações podem ser realizadas nestes dados?

Entender sobre estrutura de dados é essencial para garantir que os algoritmos sejam eficientes.

Orientação a Objetos - OO

- Método de programação (paradigma) que usa tipos de dados personalizados;
- Em vez de operar apenas com tipos de dados primitivos (inteiro, strings, booleanos, etc), podemos construir novos tipos de dados;
- Baseia-se no conceito de Objetos, criados a partir de uma estrutura chamada Classe, que são os blocos de construção dos programas/algoritmos que se utilizam deste paradigma.





Vantagens da OO

- Fornece uma estrutura modular (pequenas partes) para construção de programas;
- O software se torna mais fácil de manter;
- Reuso de código: desenvolvimento mais rápido;
- Objetos podem ser utilizados em aplicações diferentes;
- Encapsulamento: não é necessário conhecer a implementação interna de um objeto para poder usá-lo.