



Abordagens para Resolução de Problemas

UDESC - Universidade do Estado de Santa Catarina
Departamento de Ciência da Computação

CAL - Complexidade de Algoritmos

Teodoro Alberto Borges Junior

Abordagens para Resolução de Problemas

Muitos problemas podem ser resolvidos de diversas maneiras diferentes.

Contudo, dependendo da abordagem escolhida, uma solução pode ser melhor ou pior do que outra em questões de tempo ou uso de memória.

Abordagens para Resolução de Problemas

Podemos classificar os principais métodos de resolução de problemas em relação a abordagem utilizada:

- Indução matemática - fórmula
- Divisão e Conquista (*divide and conquer*)
- Algoritmos Gulosos (*greedy*)
- Algoritmo de Tentativa e Erro (*backtracking*)
- Programação Dinâmica (*dynamic programming*)
- Algoritmos de Aproximação (*approximation*)

Indução Matemática - fórmula

Abordagem simples.

O problema pode ser reduzido a uma fórmula ou a um conjunto de fórmulas matemáticas.

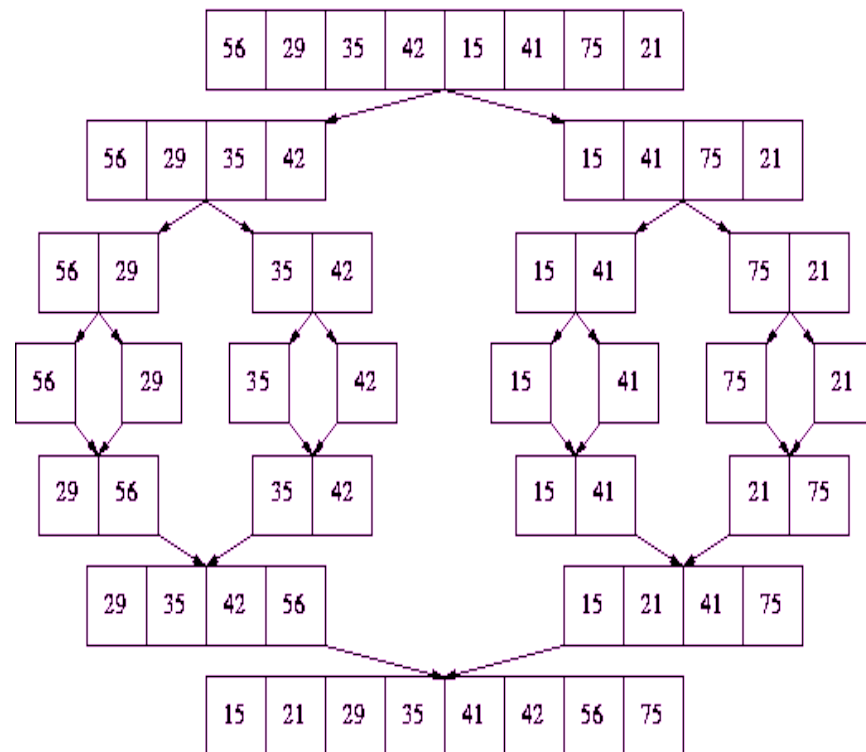
Exemplo: descobrir as raízes de uma equação de segundo grau.

Divisão e Conquista

Já foi visto em algoritmos de ordenação.

A idéia é desmembrar o problema original em vários subproblemas menores, resolvê-los e combinar as soluções.

Exemplo: Merge-Sort...



Algoritmos Gulosos – (*Greedy Algorithms*)

Uma técnica para resolver problemas de otimização: minimizar ou maximizar um determinado valor.

Um algoritmo guloso sempre faz a escolha que parece ser a melhor em um determinado momento, esperando que essa melhor escolha local leve ao melhor resultado do problema como um todo.

Algoritmos Gulosos – (*Greedy Algorithms*)

Alguns problemas que podem ser resolvidos com algoritmos gulosos:

- Problema do troco - forma canônica
- Árvore Geradora Mínima (*Minimum Spanning Tree*)
- Escalonamento de tarefas
- Problema da Mochila - fracionada (*Knapsack*)
- Codificação de Huffman

Problema do troco

Exemplo 1: considerando o conjunto de moedas disponíveis $\{1, 5, 10, 25, 50, 100\}$, qual é a quantidade mínima de moedas necessário para atender o troco de R\$ 3,82?

Problema do troco

Exemplo 1: considerando o conjunto de moedas disponíveis {1, 5, 10, 25, 50, 100}, qual é a quantidade mínima de moedas necessário para atender o troco de R\$ 3,82?

$$382 / 100 = 3 \text{ e sobram } 82$$

3 moedas de 100

$$82 / 50 = 1 \text{ e sobram } 32$$

1 moeda de 50

$$32 / 25 = 1 \text{ e sobram } 7$$

1 moeda de 25

$$7 / 5 = 1 \text{ e sobram } 2$$

1 moeda de 5

$$2 / 1 = 2 \text{ e sobra } 0$$

2 moedas de 1

Problema do troco

Exemplo 2: considerando o conjunto de moedas disponíveis $\{1, 3, 4\}$, qual é a quantidade mínima de moedas necessário para atender o troco de R\$ 0,06?

Problema do troco

Exemplo 2: considerando o conjunto de moedas disponíveis {1, 3, 4}, qual é a quantidade mínima de moedas necessário para atender o troco de R\$ 0,06?

$$6 / 4 = 1 \text{ e sobram } 2$$

1 moeda de 4

$$2 / 1 = 2 \text{ e sobra } 0$$

2 moedas de 1

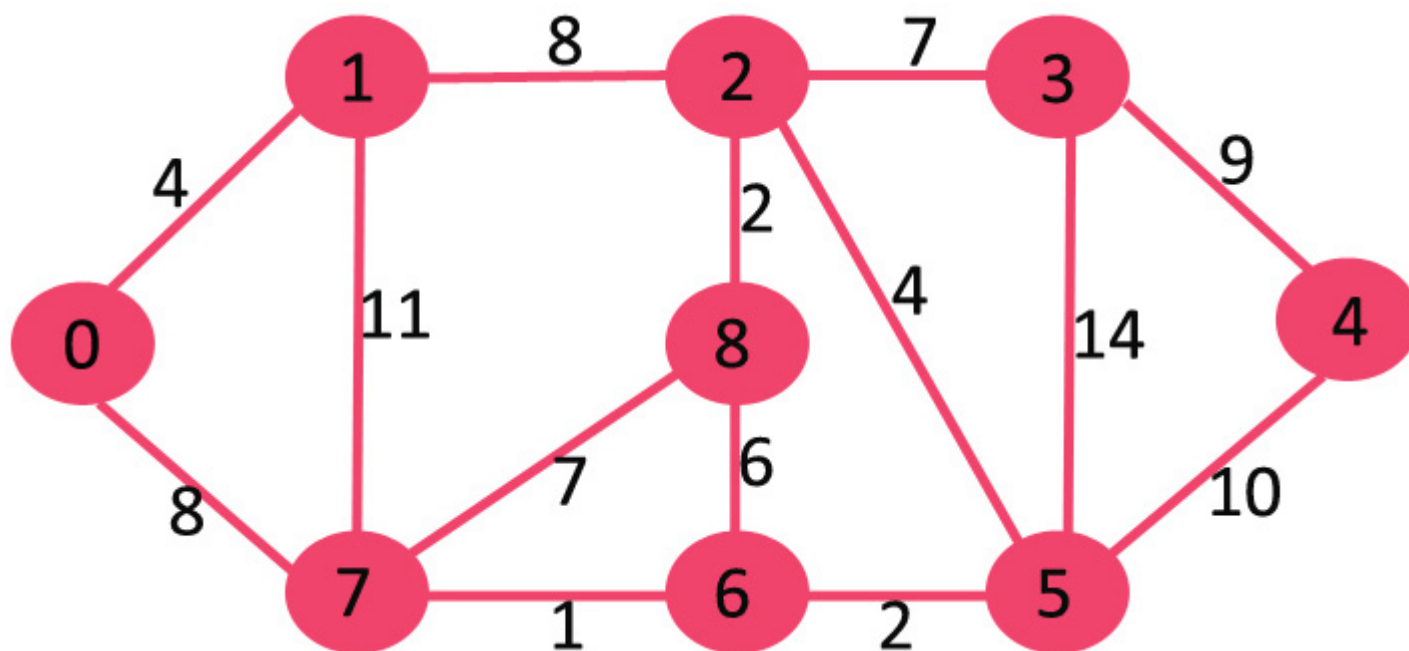
$$6 / 3 = 2 \text{ e sobra } 0$$

2 moedas de 3

Árvore Geradora Mínima

Dado um grafo com seus respectivos vértices e arestas, determinar qual a árvore que contém todos os vértices deste grafo com o menor custo possível.

Prim vs Kruskal



Problema da Mochila

Problema: um ladrão está assaltando uma loja e possui uma mochila que pode carregar um limite máximo de peso sem arrebentar. Sabendo os pesos e valores dos itens, qual é o maior valor possível que o ladrão conseguirá roubar?

Exemplo: mochila com capacidade 9 kg

| Objetos | 1 | 2 | 3 | 4 | 5 | 6 |
|---------|---|---|---|---|---|----|
| Peso | 1 | 3 | 2 | 5 | 7 | 9 |
| Valor | 3 | 2 | 4 | 7 | 9 | 12 |

Fracionada: objetos podem ser selecionados/escolhidos de forma parcial.

Problema da Mochila - fracionada

Fracionada: pode ser resolvida com algoritmo guloso calculando a proporção valor/peso.

| Objetos | 1 | 3 | 4 | 6 | 5 | 2 |
|-----------|---|---|-----|------|------|------|
| Peso | 1 | 2 | 5 | 9 | 7 | 3 |
| Valor | 3 | 4 | 7 | 12 | 9 | 2 |
| Proporção | 3 | 2 | 1.4 | 1.33 | 1.29 | 0.67 |

Codificação de Huffman

É um método de compressão de dados. Esse método usa a frequência com que cada símbolo ocorre, em uma coleção de dados, para determinar um código de tamanho variável para o símbolo.

Codificação de Huffman

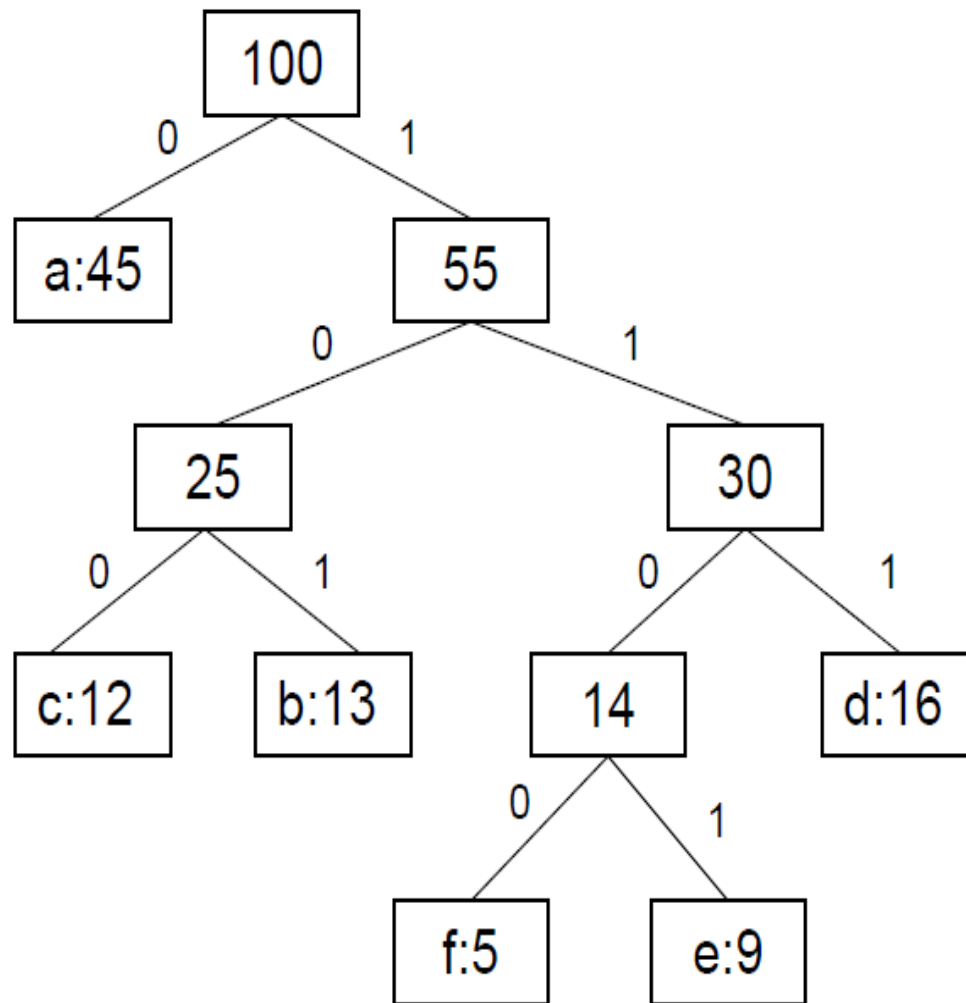
| Caractere | Frequência | Código (fixo) | Código (variável) |
|-----------|------------|------------------|----------------------|
| a | 45 | 000 | 0 |
| b | 13 | 001 | 101 |
| c | 12 | 010 | 100 |
| d | 16 | 011 | 111 |
| e | 9 | 100 | 1101 |
| f | 5 | 101 | 1100 |

... faca ...

... 101000010000 ...

... 110001000 ...

Codificação de Huffman



... faça ...

... 101000010000 ...

... 110001000 ...

Codificação de Huffman

HUFFMAN(C)

```
1   $n = |C|$ 
2   $Q = C$ 
3  for  $i = 1$  to  $n - 1$ 
4      allocate a new node  $z$ 
5       $z.left = x = \text{EXTRACT-MIN}(Q)$ 
6       $z.right = y = \text{EXTRACT-MIN}(Q)$ 
7       $z.freq = x.freq + y.freq$ 
8       $\text{INSERT}(Q, z)$ 
9  return  $\text{EXTRACT-MIN}(Q)$ 
```

Codificação de Huffman

EXERCÍCIO:

Escreva como seria codificada a sequência de caracteres “abracadabra pe de cabra” utilizando a codificação de Huffman.

Mostre a árvore criada no processo de codificação e a codificação binária com o código fixo e com o código variável.

Levar em consideração os espaços em branco.