Algoritmos Gulosos

LEONARDO VIEIRA GUIMARÃES,
MATHEUS SILVEIRA BORGES,
NARA MIRANDA DE OLIVEIRA CANGUSSU,
PABLO HENRIQUE ATAIDE OLIVEIRA,
PEDRO HUMBERTO DE ALMEIDA MENDONCA GONZAGA.

UNIMONTES

30 de Agosto de 2017

Sumário

- **Problemas de otimização são considerados um dos grandes desafios de projetos de algoritmos**, pelo fato de apresentarem um alto custo de processamento.
- Neste contexo, Cormen (2002) define que para alguns problemas de otimização, **a melhor solução é a utilização de algoritmos mais simples e mais eficientes**, como os algoritmos gulosos.
- Os **algoritmos gulosos** são aqueles que sempre fazem a **escolha que parece ser a melhor no momento**, sendo uma importante alternativa para projetos de algoritmos de otimização.

Mediante tal importância, este seminário tem como objetivo geral:

- Apresentar os **principais fundamentos** dos métodos gulosos.

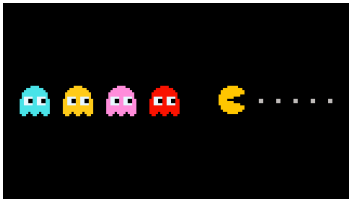
Além de ter como objetivos específicos:

- Conhecer os **problemas solúveis** pelos métodos gulosos e suas **aplicações**.
- Testar o funcionamento dos algoritmos gulosos com **implementações em Python**.

Os algoritmos gulosos apresentam as seguintes características:

- Cormen (2002) enfatiza que esse tipo de algoritmo toma as decisões que parecem **mais adequadas ao momento**.
- Estes algortimos **não reconsideram suas escolhas**, como no caso do game PacMan, que apenas come as bolinhas e nunca as volta.

Figura: Jogo clássico "Pacman". Fonte: IGN Brasil



- Usados para **resolver problemas de otimização clássicos** como o menor número de jogadas, melhor caminho do labirinto, etc.
- Ao utilizar os algoritmos gulosos, o programador espera que as **escolhas ótimas encontras para condições locais seja também escolhas ótimas a um nível global**.
- Entretanto, **as escolhas do algoritmo nem sempre expressa a solução ótima**. Não obstante, para problemas onde uma solução boa (e não necessariamente a melhor) é aceita, os algoritmos gulosos são uma importante alternativa de solução de problemas de otimização.

Os algoritmos gulosos apresentam as seguintes **vantagens** e **desvantagens**:

Vantagens:

- Algoritmos simples.
- De fácil implementação.

Desvantagens:

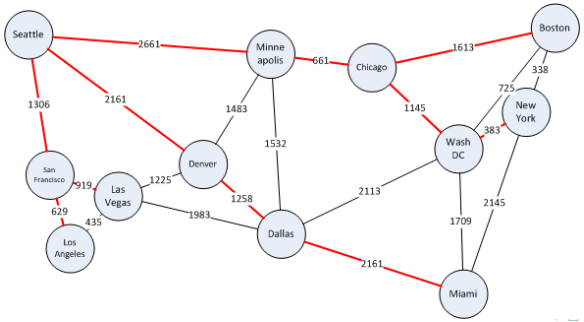
- Nem sempre conduz a solução ótimas globais.
- Podem efetuar cálculos repetitivos.

- Para construir a **solução ótima**, este algoritmo seleciona um conjunto ou **lista de candidatos de possíveis soluções.**
- A cada etapa, são acumulados um conjunto de **candidatos considerados escolhidos** e o outro de **candidatos considerados rejeitados.**
- Assim, uma função irá verificar se um conjunto particular dos candidatos selecionados **realmente produz uma solução.**
- O algoritmo dispõe também de uma função de seleção indica a **qualquer momento quais dos candidatos restantes é o mais promissor.**

- A ideia básica da estratégia gulosa é construir uma **solução "ótima"por etapas**.
- Em cada passo, após selecionar um elemento da entrada (o melhor), **decide-se se ele é viável** (caso em que virá a fazer parte da solução) **ou não**.
- **Após uma sequencia de decisões, uma solução para o problema é alcançada.**
- Nessa sequencia de decisões, **nenhum elemento é examinado mais de uma vez:** ou ele fará parte da solução, ou será descartado.

- O algoritmo de Dijkstra foi criado em 1956 por Edsger Dijkstra para **solucionar o problema do caminho mais curto.** Este algoritmo apresenta **complexidade de** $O([m+n]logn)$**.**

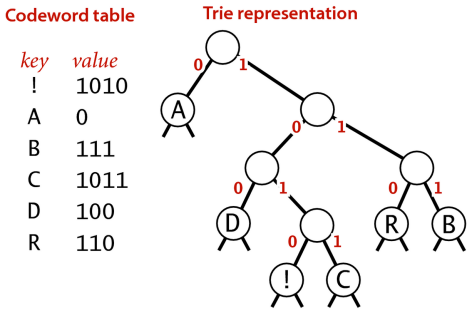
Figura: Grafo tendo cidades como vértices e estradas como arestas. Fonte: hansolav.net



- Segundo Cormen (2014, p. 141), esse algoritmo é **caracterizado como guloso** pelo fato de sempre ocorrer o relaxamento das "arestas que partem do vértice que tem o menor valor de todos os remanescentes em sua fila de prioridade".
- Seu funcionamento é bem simples: Após escolhido o **vértice inicial e o custo de referência**, é calculado o **caminho mais curto até qualquer outro vértice** obedecendo o custo mínimo referenciado.
- Este algoritmo, entretanto, **não oferece exatidão da solução se houver arcos com valores negativos**.

- O algoritmo Huffman foi desenvolvido com o objetivo de encontrar o caminho de menor custo em uma árvore binária.

Figura: Árvore binária de Huffman. Fonte: USP



- **Este algoritmo é eficiente** e amplamente utilizado em problemas de **compressão de arquivos**.
- **Cada caractere tem um código binário correspondente**, parametrizado pela sua quantidade de ocorrências no arquivo a ser comprimido.
- Assim, **um arquivo será representado por uma cadeira de caracteres binários**, dando origem o seguinte problema de otimização:

Problem

Dado um arquivo de caracteres, encontrar uma tabela de códigos que produza um arquivo codificado de comprimento mínimo.

- Sua solução é obtida convertendo um arquivo de caracteres em um arquivo de bits que **ocupe o mínimo de espaço.**
- A estratégia desta compressão consiste em **utilizar poucos bits para caracteres mais frequentes,** tornando menor o tamanho do arquivo.
- Para tal, são escolhidos **estrategicamente códigos para cada caracter** contido no arquivo.

- Vejamos um exemplo de compressão de caracteres de um arquivo com **100.000 caracteres de a até f** apresentado por Cormen (2002).

Figura: Tabela com dados do problema de compressão de caracteres. Fonte: Cormen(2002)

	a	b	c	d	e	f
Freqüência (em milhares)	45	13	12	16	9	5
Palavra de código de comprimento fixo	000	001	010	011	100	101
Palavra de código de comprimento variável	0	101	100	111	1101	1100

Figura: Árvores correspondendes ao esquema de codificação. Fonte: Cormen(2002)

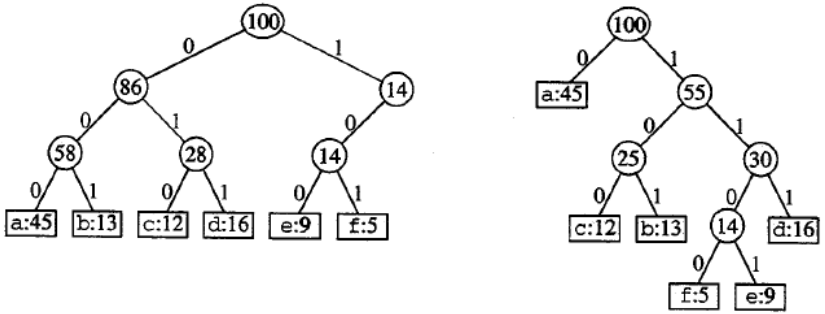
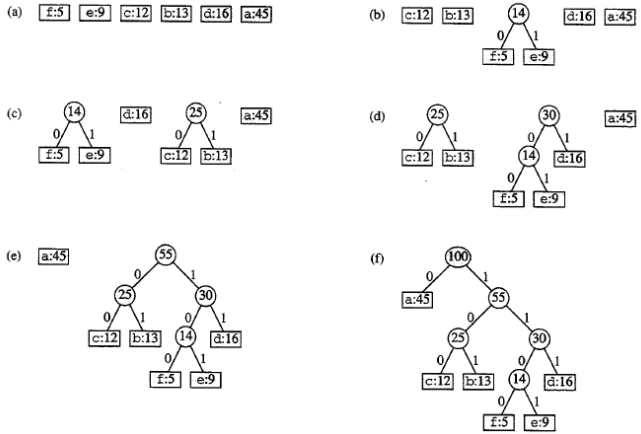


Figura: Etapas do algoritmo de Huffman. Fonte: Cormen(2002)

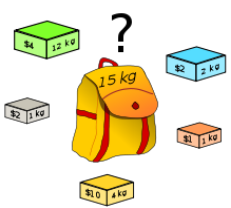


Neste presente seminário, **para melhor entendimento do conteúdo**, foram **implementados** os seguintes **algoritmos gulosos:**

- Problema da mochila
- Problema da moeda
- Problema da atividade

As implementações, a serem apresentadas a seguir, foram feitas em Python 3.6.

Figura: Problema solúveis pelos métodos gulosos. Fonte: Wikipédia, Soefcore e Informática Inteligente



Referências Bibliográficas

- CORMEN, Thomas H. *et al*. **Algoritmos:** teoria e prática. Editora Campus, v. 2, 2002.
- CORMEN, Thomas. **Desmistificando algoritmos**. Elsevier Brasil, 2014.
- FEOFILOFF, Paulo. **Algoritmos gulosos**. Disponível em: *https* : *//www.ime.usp.br / pf/analise_de_algoritmos/aulas/guloso.html*. Acesso em: 22. Ago. 2017