Projeto Olímpico de Programação

Algoritmos Gulosos

Paulo Felipe Feitosa paulofelipe.feitosa@gmail.com

Algoritmos Gulosos

- Estudamos em aulas anteriores, no tópico de **Complete Search**, alguns problemas de otimização.

- Contudo, como o próprio nome já sugere, algoritmos baseados em **Complete Search** realizam uma busca por uma parte ou por todo espaço solução.

- Porém, dependendo da dimensão dos casos de entrada do problema, nossas soluções em **Complete Search** podem não ter um bom desempenho.

- Logo, vamos estudar os algoritmos gulosos para melhorar o desempenho das nossas soluções, caso possível.

Afinal, o que são Algoritmos Gulosos?

- Problema 01: Mochila fracionária.
  - Um ladrão está dentro de uma loja com uma mochila de capacidade **C** de kilos. Dentro desta loja existem **N** produtos de peso **Pi**, de valor **Vi** e **Qi** quantidades em estoque, onde 1 <= i <= **N**.
  - O ladrão quer maximizar seu lucro na escolha dos produtos que devem ser colocados na mochila.
- Como seria uma solução em Complete Search?
- Indiquem uma solução gulosa!
- Porque é uma solução gulosa?

Afinal, o que são Algoritmos Gulosos?

- Um algoritmo guloso sempre faz a escolha que parece ser a melhor no momento em questão.
- Um algoritmo guloso é “míope”: ele toma decisões com base nas informações disponíveis na iteração corrente, sem avaliar as consequências que dessas decisões no futuro.
- Um algoritmo guloso jamais se arrepende das suas escolhas.
- A técnica gulosa nem sempre produz a melhor solução.

Afinal, o que são Algoritmos Gulosos?

- Problema 02: Problema do Troco
  - Você têm um estoque muito grande de **N** moedas de diferentes valores **M1, M2, …, Mn**.
  - Você precisa passar um troco para um cliente, mas deseja passar o mínimo de moedas possíveis. Quais e quantas moedas devem ser dadas como troco ao cliente?

- Entrada:
  - **N** = 4, **MOEDAS** = {25, 10, 5, 1} e um troco de 42 centavos, qual seria a resposta?
  - **N** = 3, **MOEDAS** = {4, 3, 1} e um troco de 6 centavos, qual seria a resposta?

Afinal, o que são Algoritmos Gulosos?

- Podemos observar que no caso do Problema 02 uma solução gulosa talvez não seja a ideal.

Algoritmos Gulosos - Problemas Clássicos

- **Balanceamento de Carga**:
  - Dadas **C** pilhas que podem empilhar 0, 1 ou 2 pratos, dados **S** <= **2C** pratos e uma lista com o tamanho, em altura, de cada prato, determine onde cada prato deve ser empilhado para minimizar o "*imbalance*".
  - Sendo **A** a média das alturas dos pratos.

$$A = \left(\sum_{j=1}^{S} Hj\right) / C$$

  - O "*imbalance*" é definido da seguinte forma:

$$Imbalance = \sum_{i=1}^{C} \left| X_i - A \right|$$

Algoritmos Gulosos - Problemas Clássicos

- **Balanceamento de Carga**:
  - Give to me solutions, now!!!
  - Vamos fazer algumas observações:
    - Observação 01: Se existe uma pilha vazia, é mais benéfico, e nunca pior, mover um prato, de uma pilha com dois pratos, para um prato vazio.
    - Observação 02: Se **S** > **C**, então S - C pilhas devem ser colocadas em pilhas que já tem pratos (Princípio da casa dos pombos).
    - Solução!? Ordenação :P
  - Aplicações?
    - Algoritmo de Balanceamento de Cargas - Algoritmo de Aproximação (Redes, Web, #Amazon).

Algoritmos Gulosos - Problemas Clássicos

- **Balanceamento de Carga**:
  - Para **S** > **2C** esse problema se torna um NP-Hard e só é resolvido com heurísticas.

Algoritmos Gulosos - Problemas Clássicos

- **Cobertura de intervalos:**
  - Dado um conjunto **S** que contém **N** intervalos encontre a quantidade mínima de intervalos necessários para cobrir o intervalo **[0, m]**, onde **m** é dado.
  - Vamos fazer algumas observações:
    - Dada a configuração já existente dos intervalos, a solução ótima seria tirar todos os intervalos que já foram cobertos?
    - Sempre teremos na solução o intervalo mais a esquerda?
    - How do we do it!?
      - Algoritmo Guloso :P

Algoritmos Gulosos - Problemas Clássicos

- **Cobertura de intervalos:**
  - Problema: 10382 - Watering Grass
-

Algoritmos Gulosos - Referências

- **Competitive Programming 3;**

- http://www.cs.yorku.ca/~andy/courses/3101/lecture-notes/IntervalCover.html;