

# Construção e Análise de Algoritmos

## lista de exercícios 25

### 1. O problema do troco

Imagine que você tem uma grande quantidade de moedas de cada um dos valores  $v_1, v_2, \dots, v_n$ .

O problema consiste em formar uma coleção de moedas cuja soma total de valores seja igual a  $S$  — você pode usar mais de uma moeda com o mesmo valor se quiser.

E o objetivo consiste em encontrar uma coleção de soma  $S$  com a menor quantidade de moedas possível.

#### a) *Estratégia gulosa*

Abaixo nós temos uma estratégia gulosa natural para esse problema

- ordene as moedas do maior valor para o menor, e  
para cada valor, selecione tantas moedas quanto possível desse valor

a1) Argumente que essa estratégia encontra uma solução ótima para o problema, quando os valores das moedas são 1, 2, 5, 10.

a2) Encontre uma coleção de valores para as moedas e um valor de  $S$ , para os quais a estratégia gulosa não encontra uma solução ótima.

#### b) *Programação dinâmica*

Apresente um algoritmo de programação dinâmica que encontra uma solução ótima para esse problema no caso geral.

Estime a complexidade do seu algoritmo em termos de  $S$  e  $n$ .

### 2. Soma de subconjuntos

Esse problema é realmente simples (de explicar ...).

Nós temos um conjunto de números inteiros  $A = \{a_1, a_2, \dots, a_n\}$  e um número  $T$ .

O problema consiste em determinar se existe algum subconjunto de  $A$  cuja soma dos elementos seja exatamente igual a  $T$ .

a) Apresente um algoritmo de programação dinâmica que resolve esse problema.

b) Explique porque o seu algoritmo pode ser utilizado para resolver o problema da partição de conjuntos (ver aula 15).

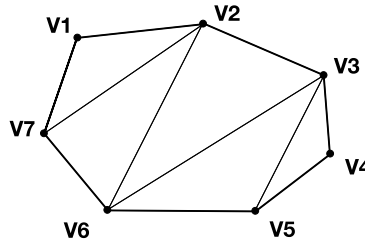
c) Explique porque isso não nos dá um algoritmo de tempo polinomial para o problema da partição de conjuntos.

### 3. Triangulação de polígonos

Considere um polígono convexo  $P$  com  $n$  vértices  $V_1, V_2, \dots, V_n$ .

Uma triangulação do polígono  $P$  consiste em uma coleção de linhas que ligam os seus vértices, mas que não se intersectam, e que dividem o polígono em triângulos.

A figura abaixo apresenta um exemplo de triangulação de um polígono.



Para cada par de vértices  $V_i, V_j$  existe um peso  $p_{ij}$  associado à linha que liga esses dois vértices.

Triangulações diferentes utilizam um conjunto de linhas diferente, e podem ter um peso total diferente.

O problema consiste em encontrar uma triangulação do polígono  $P$  cujo peso total é o menor possível.

- Apresente um algoritmo de programação dinâmica para esse problema.
- Estime a complexidade do seu algoritmo.

### 4. O problema do estoque

Imagine que você está no ramo das vendas de tratores.

Seus clientes fazem os pedidos com bastante antecedência, e você já sabe quantas entregas você terá que fazer nos próximos  $n$  meses

$$e_1, e_2, e_3, \dots, e_n$$

As entregas sempre acontecem no início do mês, e os tratores que não são vendidos ficam guardados em um depósito, a um custo  $C$  por trator por mês.

Além disso, o fornecedor de tratores cobra uma taxa  $T$  por cada remessa de tratores que ele envia para você (além do preço dos tratores, é claro).

Quer dizer, por um lado você gostaria de fazer remessas grandes para economizar nas taxas, mas por outro lado você gostaria de fazer remessas pequenas para economizar o custo de armazenamento.

Finalmente, para complicar as coisas um pouquinho mais, o depósito que você utiliza tem uma capacidade máxima para  $L$  tratores.

O problema consiste em definir uma sequência de remessas a serem feitas ao fornecedor

$$r_1, r_2, \dots, r_k$$

onde cada remessa  $r_j$  indica o mês e a quantidade de tratores solicitada ao fornecedor. (Imagine que o fornecedor entrega os tratores imediatamente a você.)

As remessas devem ser suficientes para realizar todas as entregas de cada mês, e você não pode exceder a capacidade de armazenamento do depósito.

O objetivo é encontrar uma sequência de remessas que minimize o custo total com o armazenamento e as taxas.

- a) Apresente um algoritmo de programação dinâmica para esse problema.
- b) Estime a complexidade do seu algoritmo.