AULA 17

Algoritmos gulosos (*greedy*)

CLRS 16.2

Máximo e maximal

$\mathcal{S}$ = coleção de subconjuntos de $\{1, \ldots, n\}$

Elemento X de $\mathcal{S}$ é **máximo** se
não existe Y em $\mathcal{S}$ tal que $|Y| > |X|$.

Elemento X de $\mathcal{S}$ é **maximal** se
não existe Y em $\mathcal{S}$ tal que $Y \supset X$.

Exemplo:
$\{\{1,2\},\{2,3\},\{4,5\},\{1,2,3\},\{2,3,4,5\},\{1,3,4,5\}\}$

$\{1,2\}$ não é maximal
$\{1,2,3\}$ é maximal
$\{2,3,4,5\}$ é maximal e máximo

Problema 1: Encontrar elemento máximo de $\mathcal{S}$.
Usualmente difícil.

Problema 2: Encontrar elemento maximal de $\mathcal{S}$.
Muito fácil: aplique algoritmo "guloso".

$\mathcal{S}$ "tem estrutura gulosa" se todo maximal é máximo.
Se $\mathcal{S}$ tem estrutura gulosa, Problema 1 é fácil.

Algoritmos gulosos

Algoritmo guloso

- procura maximal e acaba obtendo máximo
- procura ótimo local e acaba obtendo ótimo global

costuma ser

- muito simples e intuitivo
- muito eficiente
- difícil provar que está correto

Problema precisa ter

- subestrutura ótima
  (como na programação dinâmica)
- propriedade da escolha gulosa
  (*greedy-choice property*)

Variante da soma de subconjunto

Dados inteiros não-negativos $w_1, \ldots, w_n$ e W, encontrar um subconjunto máximo K de $\{1, \ldots, n\}$ dentre os que satisfazem

$$\textstyle\sum_{k \in K} w_k \leq W \ .$$

Subestrutura ótima (já vimos):

Se K é ótimo para (n, W) e $n \in K$
então $K - \{n\}$ é ótimo para $(n-1, W - w_n)$
senão K é ótimo para $(n-1, W)$.

Propriedade da escolha gulosa:

Se $w_n \leq w_i$ para todo i e $w_n \leq W$
então n pertence a alguma solução ótima.

Prove as propriedades!

Simplificação: basta dar *tamanho* de K ótimo

Algoritmo guloso, versão recursiva,
supondo $w_1 \geq \cdots \geq w_n$:

```
SS-REC (w, n, W)
  se n = 0 ou w_n > W
      então devolva 0
      senão devolva 1 + SS-REC (w, n − 1, W − w_n)
```

Consumo de tempo: $T(n) = T(n-1) + \mathrm{O}(1)$

Consumo de tempo: $\mathrm{O}(n)$

Algoritmo guloso, versão iterativa,
supondo $w_1 \geq \cdots \geq w_n$:

```
SS-ITER (w, n, W)
soma ← 0
para j ← n decrescendo até 1 faça
    se w_j > W
        então devolva soma
        senão soma ← soma + 1
              W ← W − w_j
devolva soma
```

Prove que o algoritmo está correto!

Consumo de tempo: $O(n)$

Prog din consumiria $\Theta(n^2)$

TAREFA 17

Exercício 17.A
O problema da soma de subconjunto do exercício 15.F pode ser resolvido por um algoritmo guloso? O problema tem a propridade da escolha gulosa?

Exercício 17.B
O problema da mochila 0-1 (exercício 16.G) pode ser resolvido por um algoritmo guloso?

Exercício 17.C [Mochila fracionária. CLRS 16.2-1]
O problema da mochila fracionária consiste no seguinte: dados números inteiros não-negativos $v_1, \ldots, v_n$ e $w_1, \ldots, w_n, W$, encontrar racionais $x_1, \ldots, x_n$ no intervalo $[0, 1]$ tais que

$$x_1 w_1 + \cdots + x_n w_n \leq W \quad \text{e} \quad x_1 v_1 + \cdots + x_n v_n \text{ é máxima.}$$

(Imagine que w_i é o *peso* e v_i é o *valor* do objeto i.) Escreva um algorito guloso para resolver o problema.

Para provar que o seu algoritmo está correto, verifique as seguintes propriedades. Propriedades da subestrutura ótima: se $x_1, .., x_n$ é solução ótima do problema (n, W) então $x_1, .., x_{n-1}$ é solução ótima do problema $(n-1, W - x_n w_n)$. Propriedade da escolha gulosa: se $v_n/w_n \geq v_i/w_i$ para todo i então existe uma solução ótima x tal que $x_n = \min(1, W/w_n)$.

AULA 19

Mais algoritmos gulosos

CLRS 16.2