

MO417 – Ata do Exercício 16.2-2

Alexandre Toshio Hirata

30 de abril de 2010

Enunciado: Dê uma solução em programação dinâmica para o problema da mochila 0-1 que roda em tempo $O(nW)$, onde n é o número de itens e W é o peso máximo de itens que o ladrão pode colocar em sua mochila.

Seja v_i o valor do item i e w_i o peso do item i . Primeiramente, observe que o problema exige que os pesos sejam inteiros.

Definindo $z[i, w]$ como sendo o máximo valor possível escolhendo entre os itens $1, \dots, i$ numa mochila com peso máximo w (para $w = 1..W$), podemos definir uma relação recursiva para z .

Se não há itens a serem colocados na mochila ou se esta não suporta nenhum peso, a solução é trivial. Caso o acréscimo do peso do i -ésimo elemento ultrapasse o limite suportado pela mochila, então a solução ótima até o momento é a situação corrente. Caso suporte, então a solução é dada pelo máximo entre a solução corrente e a solução em que adicionamos o item i à mochila. Assim, temos que z pode ser calculada assim:

$$z[i, w] = \begin{cases} 0, & \text{se } n = 0 \vee w = 0 \\ z[i - 1, w], & \text{se } w_i > w \\ \max(z[i - 1, w - w_i] + v_i, z[i - 1, w]) & \text{se } i > 0 \wedge w \geq w_i \end{cases} \quad (1)$$

A seguir, temos o Algoritmo 1 - KNAPSACK, escrito pelo aluno Pedro Hokama e validado pelos alunos em sala e pelo monitor que implementa a solução do problema em questão. Ele recebe como entrada os vetores $v[1..n]$ e $w[1..n]$ e o peso máximo da mochila W . A matriz K equivale à nossa z , porém os índices são ligeiramente diferentes (basta observar que a solução encontra-se em $K[1, W]$), mas isto não invalida o algoritmo.

Algorithm 1 KNAPSACK($v[1..n], w[1..n], W$)

```
 $K[1..n + 1, 0..W]$   
for  $w \leftarrow 0$  to  $W$  do  
     $K[n + 1, w] \leftarrow 0$   
end for  
for  $i \leftarrow n$  downto 1 do  
    for  $w \leftarrow 1$  to  $W$  do  
        if  $w[i] \leq w$  then  
             $K[i, w] \leftarrow \max(K[i + 1, w - w[i]] + v[i], K[i + 1, w])$   
        else  
             $K[i, w] \leftarrow K[i + 1, w]$   
        end if  
    end for  
end for  
return  $K[1, W]$ 
```
