Minimo de Vetor

O Problema de **Mínimo de Vetor**

Mínimo de Vetor

Instância: (v, a, b), onde v é um vetor indexado por $[a..b]$, com $a \leq b$.
Resposta: $m \in [a..b]$ tal que $v[m] \leq v[i]$ para todo $i \in [a..b]$.

Minimo(v, a, b)

```
m ← i ← a
Enquanto i < b
    i ← i + 1
    Se v[i] < v[m]
        m ← i
Devolva m
```

e concluímos que

$$\text{Minimo}(v, a, b) \text{ efetua } b - a \text{ comparações entre elementos de } v.$$

Solução Recursiva

Poderíamos escrever

Minimo'(v, a, b)

```
m ← solução da instância (v, a, b − 1)
Se v[b] < v[m]
    m ← b
Devolva m
```

ou seja

Minimo'(v, a, b)

```
m ← Mínimo′(v, a, b − 1)
Se v[b] < v[m]
    m ← b
Devolva m
```

Exemplo. *Executar Mínimo*$'(v, 1, 2)$ *para*

i	1	2	3	4	5	6	7
$v[i]$	16	23	4	42	15	8	4

Quando $a = b$, a tripla $(v, a, b-1)$ não é uma instância do problema de Mínimo de Vetor. Então,

```
Minimo'(v, a, b)
Se a = b
    Devolva a
m ← Minimo'(v, a, b − 1)
Se v[b] < v[m]
    m ← b
Devolva m
```

Exemplo. *Executar Minimo*$'(v, 1, 3)$ *e Minimo*$'(v, 4, 7)$.

Um *algoritmo recursivo* é um algoritmo que invoca a si próprio.
Num algoritmo recursivo existem **sempre** 3 elementos:

Base: Computação das instâncias que não são resolvidas de maneira recursiva;

Recursão: Invocação do algoritmo para outra instância do problema;

Passo: Transformação da resposta dessa outra instância em uma resposta da instância original.

No caso do algoritmo Minimo$'$ temos que

Base: as instâncias (v, a, b) onde $a = b$.

Recursão: a computação de Minimo$'(v, a, b-1)$.

Passo: a transformação da resposta "parcial" de Minimo$'(v, a, b-1)$ numa resposta "completa" de Minimo$'(v, a, b)$.

Análise

Quantas comparações entre elementos de v são feitas na execução de Minimo$'(v, a, b)$?

O número n de elementos no vetor de uma instância (v, a, b) do problema é $b - a + 1$, ou seja $n = b - a + 1$.

Fazendo

$C'(n) =$ número de comparações na execução de $\text{Minimo}'(v,a,b)$ quando $b-a+1=n$,

temos

$$C'(n) = \begin{cases} 0, & \textsf{se } n=1, \\ 1 + \text{número de comparações na execução de Minimo}'(v,a,b-1), & \textsf{se } n>1. \end{cases}$$

Como na execução de $\text{Minimo}'(v,a,b-1)$ temos

$$(b-1)-a+1 = (b-a+1)-1 = n-1$$

elementos, então

$$\text{número de comparações na execução de Minimo}'(v,a,b-1) = C'(n-1),$$

e

$$C'(n) = \begin{cases} 0, & \textsf{se } n=1, \\ 1 + C'(n-1), & \textsf{se } n>1. \end{cases}$$

Recorrências

C' é uma função definida recursivamente, ou seja, uma *recorrência*.

Exemplo. *Calcular* $C'(7)$.

Em geral, a análise de um algoritmo recursivo resulta numa recorrência.

Chama-se *solução de uma recorrência* a expressão não recursiva de uma recorrência.

Solução de recorrências é um dos assuntos de Matemática Discreta e Análise de Algoritmos. Nesta disciplina vamos falar disso muito superficialmente.

A recorrência

$$C'(n) = \begin{cases} 0, & \textsf{se } n=1, \\ 1 + C'(n-1), & \textsf{se } n>1, \end{cases}$$

é bastante simples pois para todo $n>0$ temos

$$\begin{aligned} C'(n) &= 1 + C'(n-1) \\ &= 1 + (1 + C'((n-1)-1)) = 2 + C'(n-2) \\ &= 2 + (1 + C'((n-2)-1)) = 3 + C'(n-3) \\ &= \dots \\ &= (n-1) + C'(n-(n-1)) = (n-1) + C'(1) \\ &= n-1. \end{aligned}$$

Chão e Teto

Definição. *Dado* $x \in \mathbb{R}$*, o* chão *de* x *é o maior inteiro menor ou igual a* x*, ou seja,*

$$\lfloor x \rfloor = \max \{z \in \mathbb{Z} \mid z \leq x\},$$

e o teto *de* x *é o menor inteiro maior ou igual a* x*, ou seja,*

$$\lceil x \rceil = \min \{z \in \mathbb{Z} \mid z \geq x\},$$

Exemplo.

$$\begin{aligned}
\lfloor 2 \rfloor &= 2; \\
\lceil 2 \rceil &= 2; \\
\lfloor z \rfloor &= z, \text{ para todo } z \in \mathbb{Z}; \\
\lceil z \rceil &= z, \text{ para todo } z \in \mathbb{Z}; \\
\left\lfloor \frac{35}{23} \right\rfloor &= 1; \\
\left\lceil \frac{35}{23} \right\rceil &= 2; \\
\left\lfloor \frac{-35}{23} \right\rfloor &= -2 \\
\left\lceil \frac{-35}{23} \right\rceil &= -1.
\end{aligned}$$