

1. TÉCNICA DE DIVISÃO-E-CONQUISTA:

Dada uma instância para um problema:

- a) Se ela for trivial, resolva-a diretamente.
- b) Se ela não for trivial:
 - i) DIVISÃO: divida-a em instâncias menores.
 - ii) CONQUISTA: resolva essas instâncias RECURSIVAMENTE (isto é, pela mesma estratégia de divisão-e-conquista).
 - iii) COMBINAÇÃO: combine as soluções das instâncias menores, de forma a obter uma solução para a instância original.

2. EXEMPLO DA TÉCNICA: ordenação por entrelaçamento ("mergesort").

```
=====
Algoritmo: ord_entrel_casca
Entrada: vetor A[1..n] de números
Saída: nenhuma
```

- ```

1. obter vetor auxiliar B[1..n]
2. ord_entrel(A,B,1,n)
=====
```

```
=====
Algoritmo: ord_entrel
Entrada: vetores A[1..n] e B[1..n] e índices "i" e "f"
Saída: nenhuma
```

- ```
-----
1. SE i < f
2. | m := piso((i+f)/2)
3. | ord_entrel(A,B,i,m)
4. | ord_entrel(A,B,m+1,f)
5. | entrelaçar(A,B,i,m,f)
=====
```

```
=====
Algoritmo: entrelaçar
Entrada: vetores A[1..n] e B[1..n] e índices "i", "m" e "f"
Saída: nenhuma
```

- ```

01. PARA k de i a f
02. | B[k] := A[k]
03. a := i, b := m+1, k := i
04. ENQUANTO verdadeiro
05. | SE B[a] <= B[b]
06. | | A[k] := B[a]
07. | | ++k, ++a
08. | | SE a > m
09. | | | REPITA
10. | | | A[k] := B[b]
11. | | | ++k, ++b
12. | | ENQUANTO b <= f
13. | | RETORNE.
14. | SENÃO
15. | | A[k] := B[b]
16. | | ++k, ++b
17. | | SE b > f
18. | | | REPITA
19. | | | A[k] := B[a]
20. | | | ++k, ++a
21. | | ENQUANTO a <= m
```

22. | | | RETORNE.

=====

3. CORREÇÃO do algoritmo:

a) "f-k" é um variante do laço da linha 4 de "entrelaçar".  
(Em particular, observe que, para toda iteração que efetivamente chega ao fim,  
sempre temos  $f-k \geq 0$  ao fim da iteração.)  
Logo, como também é possível mostrar que toda iteração do laço em questão termina,  
então o laço em questão sempre termina.  
De fato, como f-k é um variante do laço, então podemos concluir que o laço executa no máximo  
 $f-i+1$  iterações, correspondentes aos valores possíveis para o variante em cada iteração:  
 $f-i, f-(i+1), f-(i+2), \dots, f-(f-1), f-f$ .

b) Invariantes do laço em questão:

\*  $i \leq k < f, i \leq a \leq m, m < b \leq f$ .  
\* Se  $k > i$ , então  $A[i..k-1]$  contém  $B[i..a-1] \cup B[m+1..b-1]$  em ordem crescente.  
\* Se  $k > i$ , então  $A[k-1] \leq B[a], B[b]$ .

c) Correção de ord\_entrel: por indução em "f-i+1", pressupondo a correção do algoritmo "entrelaçar".

4. TEMPO DE EXECUÇÃO do algoritmo ord\_entrel:

O tempo de execução do algoritmo pode ser limitado superiormente pela seguinte função:

$t(1) = a$ , para algum  $a > 0$   
 $t(n) \leq t(\text{teto}(n/2)) + t(\text{piso}(n/2)) + c*n$ , para algum  $c > 0$ .

Nós podemos também limitar esse tempo inferiormente, de forma análoga:

$t'(1) = t(1) = a$   
 $t'(n) \geq t(\text{teto}(n/2)) + t(\text{piso}(n/2)) + c'*n$ , para algum  $c' > 0$ .

5. EXERCÍCIO: mostre que  $t = O(n * \lg n)$  e  $t' = \Omega(n * \lg n)$ .