

Construção e Análise de Algoritmos

discussão 21b: Implementação iterativa

As técnicas de Divisão e Conquista e Algoritmos Gulosos são *métodos de decomposição de problemas*.

Mas, algumas pessoas preferem ver a Programação Dinâmica de outro modo: como um *método de composição de soluções*.

Quando essas pessoas olham, por exemplo, para o pseudo-código

```
Procedimento  Ativ-Pri-PD5.0 ( A[i..n] : conj. atividades com prioridades
{
    ordenadas por tempo de término )
1.   Se ( i = n+1 )
2.   { T[n+1] <-- (vazio,0);   Retorna  }

3.   Se ( T[i+1] = vazio )  Ativ-Pri-PD5.0 ( A[i+1..n] )
4.   (S1,P1) <-- T[i+1]

5.   ci <-- índice da próxima atividade compatível com ai
6.   Se ( T[i+1] = vazio )  Ativ-Pri-PD5.0 ( A[i+1..n] )
7.   S2 <-- T[ci].S + {ai};   P2 <-- T[ci].P + pi

8.   Se ( P1 > P2 )   T[i] <-- (S1,P1)
9.   Senão           T[i] <-- (S2,P2)
}
```

elas ignoram as chamadas recursivas e pensam: “*Ah, então é só isso ...*”.

Quer dizer, concentrando a atenção nas linhas 1,2,4,7,8,9, elas observam que

- a posição $T[n+1]$ sempre recebe o valor $(\emptyset, 0)$
- a posição $T[n]$ recebe o valor

$$\text{Max} \left\{ T[n+1], (\{a_n\}, p_n) + T[c(n)] \right\}$$

- a posição $T[n-1]$ recebe o valor

$$\text{Max} \left\{ T[n], (\{a_{n-1}\}, p_{n-1}) + T[c(n-1)] \right\}$$

- e assim por diante ...

Em outras palavras, elas descobrem que

- cada posição da tabela pode ser preenchida examinando os valores de posições que vem depois dela
- a última posição $T[n+1]$ sempre recebe o valor $(\emptyset, 0)$

Isso significa que a tabela $T[1..n+1]$ pode ser preenchida de trás para frente, por meio de um laço simples

```
Procedimento  Ativ-Pri-PD6.0 ( A[1..n] : conj. atividades com prioridades
{                                     ordenadas por tempo de término )
  T[1..n+1]  <--  tabela vazia

  T[n+1]  <--  (vazio,0)

  Para  i <-- n  Até  1
  {
    ci  <--  índice da próxima atividade compatível com ai

    T[i]  <--  Max { T[i+1] , ({ai},pi) + T[ci] }
  }

  Retorna ( T[1] )
}
```

Algumas pessoas preferem fazer as coisas assim ...