

Análise de Algoritmos

Projeto de Algoritmos por Indução

Projeto de Algoritmos por Indução

- A formulação do algoritmo vai ser análoga ao desenvolvimento de uma demonstração por indução.
- Assim, para resolver um problema P :
 - mostramos como resolver instâncias pequenas de P (casos base) e
 - mostramos como obter uma solução de uma instância de P a partir das soluções de instâncias menores de P

Projeto de Algoritmos por Indução

- Este processo indutivo resulta em algoritmos recursivos, em que:
 - a base da indução corresponde à resolução dos casos base da recursão,
 - a aplicação da hipótese de indução corresponde a uma ou mais chamadas recursivas e
 - o passo da indução corresponde ao processo de obtenção da resposta para o problema original a partir das respostas devolvidas pelas chamadas recursivas.

Projeto de Algoritmos por Indução

- Um benefício imediato é que o uso (correto) da técnica nos dá uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.
- Muitas vezes é imediato como converter o algoritmo recursivo em um iterativo.
- Frequentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece.

Cálculo de polinômios

- Problema: Dados uma sequência de números reais $a_n, a_{n-1}, \dots, a_1, a_0$, e um número real x , calcular o valor do polinômio

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0.$$

- Naturalmente este é um problema bem simples. Estamos interessados em projetar um algoritmo que faça o menor número de operações aritméticas (multiplicações, principalmente).

Cálculo de polinômios – solução 1

- Hipótese de indução: (primeira tentativa)
 - Dados uma sequência de números reais $a_{n-1}, \dots, a_1, a_0$, e um número real x , sabemos calcular o valor de
$$P_{n-1}(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0.$$
 - Caso base: $n = 0$. A solução é a_0 .
 - Para calcular $P_n(x)$, basta calcular x^n , multiplicar o resultado por a_n e somar o resultado com $P_{n-1}(x)$.

Algoritmo CálculoPolinomio

- Entrada: coeficientes $A = \{a_n, a_{n-1}, \dots, a_1, a_0\}$ e X
- Saída: o valor de $P_n(x)$
 1. se $n=0$ então $P \leftarrow a_0$
 2. senão
 3. $A' \leftarrow a_{n-1}, \dots, a_1, a_0$
 4. $P' \leftarrow \text{CálculoPolinomio}(A', n-1, X)$
 5. $x_n \leftarrow X$
 6. para $i \leftarrow 2$ até n faça
 7. $x_n \leftarrow x_n * X$
 8. $P \leftarrow P' + a_n * x_n$
 9. devolva(P)

Cálculo de polinômios – solução 1

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + n \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

Não é difícil ver que

$$\begin{aligned} T(n) &= \sum_{i=1}^n [i \text{ multiplicações} + 1 \text{ adição}] \\ &= n(n+1)/2 \text{ multiplicações} + n \text{ adições.} \end{aligned}$$

Observação: Essa solução perde muito tempo recalculando x^n . É possível melhorar??

Cálculo de polinômios – solução 2

- Alternativa: eliminar essa computação desnecessária trazendo o cálculo de x^{n-1} para dentro da hipótese de indução.
- Hipótese de Indução reforçada: Sabemos calcular o valor de

$$P_{n-1}(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0 \text{ e também de } x^{n-1}$$

Cálculo de polinômios – solução 2

- Então, no passo de indução, primeiro calculamos x^n fazendo $x^n \leftarrow x * x^{n-1}$, conforme exigido pela hipótese. Em seguida, calculamos $P_n(x)$ multiplicando x^n por a_n e somando o valor obtido com $P_{n-1}(x)$.
- Note que para o caso base $n = 0$, a solução agora é $(a_0, 1)$.

Algoritmo CálculoPolinomio

Entrada: coeficientes $A = \{a_n, a_{n-1}, \dots, a_1, a_0\}$ e X

Saída: o valor de $P_n(x)$ e x^n

1. se $n=0$ então $P \leftarrow a_0$ e $x_n \leftarrow 1$
2. senão
3. $A' \leftarrow a_{n-1}, \dots, a_1, a_0$
4. $(P', x') \leftarrow \text{CálculoPolinomio}(A', n-1, X)$
5. $x_n \leftarrow X * x'$
6. $P \leftarrow P' + a_n * x_n$
7. devolva(P)

Cálculo de polinômios – solução 2

Se $T(n)$ é o número de operações aritméticas realizadas pelo algoritmo, então temos a recorrência:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 2 \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

A solução da recorrência é

$$\begin{aligned} T(n) &= \sum_{i=1}^n (2 \text{ multiplicações} + 1 \text{ adição}) \\ &= 2n \text{ multiplicações} + n \text{ adições.} \end{aligned}$$

Cálculo de polinômios – solução 3

- Hipótese de indução mais reforçada:
 - Sabemos calcular o polinômio

$$P'_{n-1}(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \dots + a_1$$

Algoritmo CálculoPolinomio

Entrada: coeficientes $A = \{a_n, a_{n-1}, \dots, a_1, a_0\}$ e X

Saída: o valor de $P_n(x)$

1. se $n=0$ então $P \leftarrow a_0$
2. senão
3. $A' \leftarrow a_n, \dots, a_1$
4. $P' \leftarrow \text{CálculoPolinomio}(A', n-1, X)$
5. $P \leftarrow x * P' + a_0$
6. devolva(P)

Cálculo de polinômios – solução 3

Chamando de $T(n)$ o número de operações aritméticas realizadas pelo algoritmo, temos a recorrência:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 1 \text{ multiplicação} + 1 \text{ adição}, & n > 0. \end{cases}$$

A solução é

$$\begin{aligned} T(n) &= \sum_{i=1}^n (1 \text{ multiplicação} + 1 \text{ adição}) \\ &= n \text{ multiplicações} + n \text{ adições.} \end{aligned}$$

Essa forma de calcular $P_n(x)$ é chamada de **regra de Horner**.