

Prova de CT-234

1. Seja $P(x)$ um polinômio de grau n dado por $\sum_{k=0}^n a_k x^{n-k}$. Suponha que todos os coeficientes a_i 's são números inteiros, assim como x . Se você fizer o cálculo de $P(x)$ usando o algoritmo abaixo, quantas multiplicações serão feitas? Idem para a quantidade de somas.

Polinômio(x)

Input: vetor de coeficientes a_k , grau n do polinômio e número x .

Output: valor de $P(x)$.

```

begin
  v ← 0;
  for i ← 0 to n do
    begin
      um ← 1;
      for j ← 0 to n - i do
        um ← um.x;
      v ← v + ai.um;
    end;
  retorne v;
end.
```

2. O algoritmo de Horner, para o cálculo de $P(x)$ como acima é dado como abaixo. Determine (i) quantas multiplicações são feitas assim como (ii) quantas somas. (iii-x) Apresente uma função recursiva para o cálculo de $P(x)$ usando o algoritmo de Horner.

$$P(x) = a_n + x(a_{n-1} + x(a_{n-2} + \cdots + x(a_2 + x(a_1 + xa_0)) \cdots)).$$

3. Mostre que $P(x) = \sum_{k=0}^n a_{n-k} x^k$, para x inteiro, pode ser encontrado em tempo menor que o algoritmo de Horner se $a_0 = a_1 = \cdots = a_n$ e inteiros positivos. (Justifique sua resposta usando indução).
4. Apresente (i-iv) um algoritmo recursivo para resolver o problema das Torres de Hanói tal que se um disco não estiver na haste central, que necessite ir da haste inicial para a haste final ou da haste final para a haste inicial, então ele deverá passar antes pela haste central. (v) Qual a menor quantidade de movimentos que resolve este problema com n discos? (Justifique sua resposta usando indução).
5. Apresente (i) um algoritmo recursivo para resolver o problema de Josefo e (ii) um algoritmo iterativo para o mesmo problema.
6. Suponha que $J(n)$ seja o número associado à pessoa que irá sobreviver no problema de Josefo e que $2^{21} < n \leq 2^{42}$. Nestas condições: (i) é possível afirmar que $J^3(n) = J(J(J(n)))$ é maior do que 2^7 ? (ii-v) Faça uma estimativa para o valor de $J^3(n) = J(J(J(n)))$ caso $2^{51} + 2^{50} + 2^{41} + 2^{40} + 2^{31} + 2^{30} < n \leq 2^{51} + 2^{50} + 2^{41} + 2^{40} + 2^{31} + 2^{30} + 2^{21}$. (Justifique)

Resolução da Prova de CT-234

1 Questão 1

1. Cálculo da Quantidade de Multiplicações

Ocorrem multiplicações nas linhas 7 ($um \leftarrow um.x$) e 8 ($v \leftarrow v + a_i.um;$) do algoritmo.

O número de multiplicações na linha 7 é dado por:

$$M_1 = \sum_{i=0}^n \sum_{j=0}^{n-i} 1 \quad (1)$$

Sabemos que $\sum_{k=0}^n 1 = n$. Portanto, $\sum_{j=0}^{n-i} 1 = n - i$. Substituindo na equação acima, temos:

$$M_1 = \sum_{i=0}^n n - i \quad (2)$$

Sabemos que $\sum_{i=0}^n n - i = \sum_{i=0}^n i$ (basta notar que as somas possuem os mesmos termos em ordem invertida). Portanto:

$$M_1 = \sum_{i=0}^n i \quad (3)$$

Sabemos também que $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ (equação de Gauss). Portanto:

$$M_1 = \frac{n(n+1)}{2} \quad (4)$$

O número de multiplicações na linha 8 é dado por:

$$M_2 = \sum_{i=0}^n 1 = n + 1 \quad (5)$$

Portanto, a quantidade de multiplicações é dada por:

$$M = M_1 + M_2 = \frac{n(n+1)}{2} + n + 1 = \frac{n^2 + n + 2n + 2}{2} = \frac{n^2 + 3n + 2}{2} \quad (6)$$

2. Cálculo da Quantidade de Somas

Ocorrem somas apenas na linha 8 ($v \leftarrow v + a_i.um;$) do algoritmo.

Portanto, a quantidade total de somas do algoritmo é dada por:

$$S = \sum_{i=0}^n 1 = n + 1 \quad (7)$$

2 Questão 2

Para ajudar no raciocínio, vamos tomar como exemplo $n = 3$. Neste caso, temos:

$$P(x) = a_3 + x * (a_2 + x * (a_1 + x * a_0))$$

1. Cálculo da Quantidade de Multiplicações

No algoritmo de Horner ocorre uma multiplicação para cada um dos n termos, exceto o último (a_n). Portanto, a quantidade de multiplicações é dada por:

$$M = \sum_{i=0}^{n-1} 1 = n - 1 + 1 = n \quad (8)$$

2. Cálculo da Quantidade de Somas

Ocorre uma soma para cada um dos n termos, exceto o primeiro (a_0). Portanto, a quantidade de somas é dada por:

$$S = \sum_{i=1}^n 1 = n \quad (9)$$

3 Questão 3

Para ajudar no raciocínio, vamos tomar um exemplo com $n = 3$. Neste caso, temos:

$$P(x) = \sum_{k=0}^3 a_{3-k}x^k = a_3 + a_2x + a_1x^2 + a_0x^3$$

Se $a_0 = a_1 = \dots = a_n = a$ nosso exemplo fica:

$$P(x) = a + ax + ax^2 + ax^3 = a(1 + x + x^2 + x^3)$$

Ou seja, temos uma constante a multiplicando uma PG de razão x . Portanto:

$$P(x) = \sum_{k=0}^n a_{n-k}x^k = \sum_{k=0}^n ax^k = a \sum_{k=0}^n x^k \quad (10)$$

Para calcular a soma da PG fazemos:

$$S_n(x) = \sum_{k=0}^n x^k = x^0 + x^1 + \dots + x^n \quad (11)$$

Multiplicando por x , temos:

$$xS_n(x) = x^1 + x^2 + \dots + x^{n+1} \quad (12)$$

Subtraindo as duas equações acima, termo a termo, temos:

$$(1 - x)S_n(x) = x^0 - x^{n+1} \quad (13)$$

Considerando $x \neq 1$ temos:

$$S_n(x) = \frac{1 - x^{n+1}}{1 - x} \quad (14)$$

Quando $x = 1$ temos:

$$S_n(x) = \sum_{k=0}^n 1^k = \sum_{k=0}^n 1 = n + 1 \quad (15)$$

Portanto:

$$P(x) = \begin{cases} a \frac{1-x^{n+1}}{1-x} & \text{se } x \neq 1 \\ a(n+1) & \text{se } x = 1 \end{cases} \quad (16)$$

Note-se¹ que o termo x^{n+1} pode ser calculado em $O(\lg(n))$. Por exemplo, pode-se obter x^9 a partir de x com apenas 4 multiplicações: calcula-se x^2 e, em seguida, $x^4 = x^2 * x^2$, $x^8 = x^4 * x^4$ e, finalmente, $x^9 = x * x^8$.

Para n suficientemente grande e $|x| < 1$ o cálculo é feito em $O(1)$ pois temos:

$$P(x) = \frac{a}{1-x} \quad (17)$$

Em resumo, o cálculo de $P(x)$ é feito em:

- $O(n)$ quando os coeficientes são distintos;
- $O(\lg(n))$ quando os coeficientes são iguais a uma constante e $x \neq 1$;
- $O(1)$ quando os coeficientes são iguais a uma constante e $x = 1$;
- $O(1)$ quando $|x| < 1$ e $n \rightarrow \infty$.

Resta provar, por indução, que:

$$S_n(x) = \sum_{k=0}^n x^k = \frac{1-x^{n+1}}{1-x} \quad (18)$$

Caso base:

Para $n = 0$ temos, no lado esquerdo da equação:

$$S_0(x) = \sum_{k=0}^0 x^k = x^0 = 1 \quad (19)$$

No lado direito da equação, fazendo $n = 0$ temos:

$$S_0(x) = \frac{1-x^{0+1}}{1-x} = \frac{1-x}{1-x} = 1 \quad (20)$$

Portanto, a equação 18 vale para $n = 0$.

Hipótese Indutiva:

Supondo que a equação vale para n vamos provar que vale para $n + 1$.

De fato:

$$S_n(x) = \sum_{k=0}^{n+1} x^k = x^{n+1} + \sum_{k=0}^n x^k \quad (21)$$

¹Para mais detalhes, ver: Donald E. Knuth, *The Art of Computer Programming*, Volume 2: Seminumerical Algorithms, 3rd edition, ¶4.6.3 Pág. 455 (Addison-Wesley: San Francisco, 1998) ISBN 0201485419.

Usando a Hipótese Indutiva $\sum_{k=0}^n x^k = \frac{1-x^{n+1}}{1-x}$, temos:

$$S_n(x) = x^{n+1} + \frac{1-x^{n+1}}{1-x} \quad (22)$$

Desenvolvendo:

$$S_n(x) = x^{n+1} + \frac{1-x^{n+1}}{1-x} = \frac{(1-x)x^{n+1} + 1-x^{n+1}}{1-x} \quad (23)$$

$$S_n(x) = \frac{x^{n+1} - x^{n+2} + 1 - x^{n+1}}{1-x} \quad (24)$$

Portanto:

$$S_n(x) = \frac{1-x^{n+2}}{1-x} \quad (25)$$

C.Q.D.

4 Questão 4

5 Questão 5

6 Questão 6