### 

**INSTITUTO FEDERAL DE GOIÁS**

**CÂMPUS GOIÂNIA**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**ESTRUTURAS DE DADOS II**

Nome do Aluno: Luiz Antônio Rodrigues dos Santos

Data: 04/05/2018

Prof. Renan Rodrigues de Oliveira

**Indução Matemática, Loops Invariantes e Análise Assintótica**

2) Analise os algoritmos a seguir. Para cada algoritmo:

a) SomaPositivos(A,n)

1. S <- 0

2. para i <- 1 até n faça

3. se A[i] > 0

4. então s <- s + A[i]

5. devolva S

• **Descreva com suas palavras o que o algoritmo faz.**

O algoritmo calcula a soma dos números inteiros e positivos do vetor.

**• Qual a entrada do algoritmo? E a saída?**

A entrada do algoritmo é um vetor A de inteiros e um inteiro n que representa o tamanho do vetor.

A saída é soma dos elementos inteiros positivos do vetor A.

**• Mostre que o algoritmo para.**

No laço da linha 2, o valor de i é inicializado com 1, sendo incrementado a cada iteração do loop. Logo, o sua execução para quando i > n.

**• Enuncie com precisão os loops invariantes de cada algoritmo.**

No início de cada iteração do loop, a variável S corresponde a soma dos inteiros positivos do subvetor A[1 .. i-1].

**• Mostre a correção dos algoritmos através dos seus loops invariantes.**

Inicialização

Começamos mostrando que o loop invariante é válido antes da primeira iteração do loop, quando i=0. Antes da primeira iteração, o subvetor é vazio e o valor da variável S é 0. A soma dos inteiros positivos de um vetor vazio é 0. Assim, mostramos que a propriedade do loop invariante é válida na inicialização.

Manutenção

Em seguida, examinamos a segunda propriedade: a demonstração de que cada iteração mantém o loop invariante. O corpo do loop invariante realiza o teste A[i] > 0 para verificar se o valor A[i] é positivo. Se verdade, S recebe o antigo valor adicionado do valor de A[i]. Portanto, S encerra o loop com o valor da soma dos inteiros positivos do subvetor[1 .. i], que é o valor inicial de S para a próxima iteração (iteração i+1).

Término

Finalmente, examinamos o que ocorre quando o loop termina. O loop termina quando i excede n, isto é, quando i = n+1. Substituindo i por n+1 no enunciado do loop invariante, temos o subvetor A[1 .. n] que consiste no vetor inteiro. Dessa forma, ao término do loop, a variável S corresponde a soma dos inteiros positivos do vetor inteiro, o que significa que o algoritmo é correto.

• **Determine a função f(n) que caracteriza o tempo de execução de cada algoritmo.**

1

N+1

N

N (no pior caso)

1

T(n) = 3n + 3

b) Multiplica(n)

1. para i<- 1 até n faça

2. para j <- 1 até i faça

3. imprima i \* j \* n

• **Descreva com suas palavras o que o algoritmo faz.**

O algoritmo na primeira iteração do loop interno imprime a multiplicação do valor de i \* j \* n, a partir da segunda iteração a cada loop interno é mostrado na tela valor da iteração anterior somado com o valor da primeira iteração do loop interno.

**• Qual a entrada do algoritmo? E a saída?**

A entrada é um número inteiro n.

A saída é a multiplicação de i \* j \* n.

**• Mostre que o algoritmo para.**

No laço da linha 1, o valor de i é inicializado com 1, sendo incrementado a cada iteração do loop, logo, o sua execução para quando i > n. No laço interno da linha 2 o valor de j é inicializado com 1, sendo incrementado a cada iteração do loop, logo, a sua execução para quando j > i.

**• Enuncie com precisão os loops invariantes de cada algoritmo.**

No início de cada iteração do loop interno a partir da segunda iteração é impresso na tela o valor da iteração anterior somado com o valor da primeira iteração do loop interno.

**• Mostre a correção dos algoritmos através dos seus loops invariantes.**

Inicialização

Começamos mostrando que o loop invariante é válido a partir da segunda iteração do loop interno, quando i = 1 e j = 2. Antes da primeira iteração não há valores a serem impressos. Assim, mostramos que a propriedade do loop invariante é válida na inicialização.

Manutenção

Em seguida, examinamos a segunda propriedade: a demonstração de que cada iteração mantém o loop invariante. O corpo do loop invariante realiza imprime na tela i \* j\* n, sendo que, a partir da segunda iteração a cada loop interno é mostrado na tela valor da iteração anterior somado com o valor da primeira iteração. Se verdade, a cada iteração subsequente é impresso na tela o valor da iteração anterior somada com o valor da primeira iteração. Portanto, assim se mantém o loop a cada iteração

Término

Finalmente, examinamos o que ocorre quando o loop termina. O loop termina quando j excede i e i excede n, isto é, quando j = i+1 e i = n+1. Dessa forma, ao término dos loops, será impresso na tela o valor do loop anterior somado com o valor do primeiro loop, o que significa que o algoritmo é correto.

• **Determine a função f(n) que caracteriza o tempo de execução de cada algoritmo.**

N+1

N(N+1)/2

N

T(n) = (n² + 5n + 2) / 2

c) Media(V,n)

1. para i de 1 até n faça

2. soma = 0

3. para j de 1 até i faça

4. soma += V[j]

5. M[i] = soma / i

6. retorne M

• **Descreva com suas palavras o que o algoritmo faz.**

O algoritmo soma a quantidade i de elementos do vetor V e armazena a media dos i elementos no vetor M[ i ] e retorna o endereço de memória do vetor M.

**• Qual a entrada do algoritmo? E a saída?**

A entrada do algoritmo é um vetor V de inteiros e um inteiro n que representa o tamanho do vetor.

A saída o retorno do endereço de memória do vetor M.

**• Mostre que o algoritmo para.**

No laço da linha 1, o valor de i é inicializado com 1, sendo incrementado a cada iteração do loop, logo, o sua execução para quando i > n. No laço interno da linha 3 o valor de j é inicializado com 1, sendo incrementado a cada iteração do loop, logo, a sua execução para quando j > i.

**• Enuncie com precisão os loops invariantes de cada algoritmo.**

No início de cada iteração do loop externo soma = 0, o loop interno a cada iteração é armazenado a média da soma dos i elementos do vetor V no vetor M[ i ].

**• Mostre a correção dos algoritmos através dos seus loops invariantes.**

Inicialização

Começamos mostrando que o loop invariante é válido a partir da primeira iteração do loop externo com soma = 0 e i = 1, no loop interno j = 1 e é inicializado o vetor V e M. Antes da primeira iteração não há valores declarados. Assim, mostramos que a propriedade do loop invariante é válida na inicialização.

Manutenção

Em seguida, examinamos a segunda propriedade: a demonstração de que cada iteração mantém o loop invariante. O corpo do loop invariante realiza a soma da quantidade i de elementos do vetor V e armazena a media dos i elementos no vetor M[ i ]. Se verdade, a cada iteração subsequente o vetor M[ i ] faz a média e armazena os i elementos do vetor V. Portanto, assim se mantém o loop a cada interação

Término

Finalmente, examinamos o que ocorre quando o loop termina. O loop termina quando j excede i e i excede n, isto é, quando j = i+1 e i = n+1. Dessa forma, ao término dos loops, a função recebe o retorno do endereço do memória do Vetor M, o que significa que o algoritmo é correto.

• **Determine a função f(n) que caracteriza o tempo de execução de cada algoritmo.**

N+1

N

N(N+1) / 2

N(N) / 2

N

1

T(n) = 2n² + 7n + 4

d) Potência(a,n)

1. x <- 1

2. enquanto n > 0 faça

3. x <- x \* a

4. n <- n - 1

5. retorne x

• **Descreva com suas palavras o que o algoritmo faz.**

O algoritmo calcula a potência de a^n.

**• Qual a entrada do algoritmo? E a saída?**

A entrada do algoritmo os números inteiro a e n, onde a representa a base e n o expoente da potenciação.

A saída o retorno de a^n.

**• Mostre que o algoritmo para.**

No laço da linha 2, o valor de n diminui a cada iteração, logo, sua execução para quando o teste condicional for falso, ou seja, sua execução para quando n <= 0.

**• Enuncie com precisão os loops invariantes de cada algoritmo.**

No início da iteração do loop o valor de x é a^0, e a cada iteração soma-se ao valor da potenciação da iteração anterior mais 1.

**• Mostre a correção dos algoritmos através dos seus loops invariantes.**

Inicialização

Começamos mostrando que o loop invariante é válido a partir da primeira iteração do loop quando a^0 é 1. Antes da primeira iteração x = 1 que é o mesmo que a^0. Assim, mostramos que a propriedade do loop invariante é válida na inicialização.

Manutenção

Em seguida, examinamos a segunda propriedade: a demonstração de que cada iteração mantém o loop invariante. A cada iteração é somado ao valor da potenciação da iteração anterior mais 1. Se verdade, a cada iteração x recebe a potenciação de a somando mais 1. Portanto, assim se mantém o loop a cada interação

Término

Finalmente, examinamos o que ocorre quando o loop termina. O loop termina quando a condição é falsa, ou seja, quando n <= 0. Dessa forma, ao término do loop, a função recebe o retorno do endereço de x, o que significa que o algoritmo é correto.

• **Determine a função f(n) que caracteriza o tempo de execução de cada algoritmo.**

1

N+1

N

N

1

T(n) = 3n + 3

1. Coloque as funções a seguir em ordem crescente assintoticamente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 54.021 | 350n + 1 | log n + 5n - 3 | log n + 500n + 5 | n log n |
| N^2 + 2 | 5n^2 + 2^n + 1 | N^3 – 2n + 5 | 7n^5 + n | n! + 6 |

1. O que significa dizer que uma função g(n) é O(f(n)) ?

Significa que f(n) é o limite superior para g(n) e que g(n) é <= f(n).

5) O que significa dizer que um algoritmo executa em tempo proporcional a n?

Significa dizer que quanto maior o número n, maior é o tempo de execução.

6) Utilizando as definições para as notações assintóticas, prove se são verdadeiras ou falsas as seguintes afirmativas:

a) 3n³ + 2n² + n + 1 = O(n³)

3n³+ 2n² + n + 1<= cn³

3 + 2/n + 1/n2 + 1/n3 <= c

Quando n = 1, o menor valor possível de c é: 3 + 2 + 1 + 1 = 7.

Assim, a afirmação é verdadeira para c = 7 e n0 = 1.

b) 9n³ + 3n = Ω(n)

9n³ + 3n <= cn

9n² + 3 <= c

Quando n = 1, o menor valor possível de c é: 3(1)² + 3 = 12.

Assim, a afirmação é verdadeira para c = 12 e n0 = 1.

c) 5n² + 7n = Θ(n²)

c1(n²) <= 5n² + 7n <= c2(n²)

c1 <= 5 + 7/n <= c2

Quando n = infinito, o menor valor possível de c1 é: 5 + 7/infinito = 5.

Quando n = 1, o menor valor possível de c2 é: 5 + 7/1 = 12.

Assim, a afirmação é verdadeira para c1 = 5 n0 = infinito e c1 = 12 n0 = 1.