MCTA003-17 - ANÁLISE DE ALGORITMOS
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
UNIVERSIDADE FEDERAL DO ABC
PROF. MAYCON SAMBINELLI

Correção de algoritmos

1. O algoritmo *Bubble Sort* é um clássico da computação. Ele recebe um vetor $A[1..n]$ e promete ordená-lo. Sua ideia é percorrer o vetor várias vezes, trocando a ordem de pares de elementos adjacentes que estejam fora de ordem. Seu pseudocódigo é dado a seguir:

```
1: Função BUBBLESORT(A, n)
2:     Para i = n até 2, decrementando faça
3:         Para j = 1 até i − 1, incrementando faça
4:             Se A[j] > A[j + 1] então
5:                 troque A[j] com A[j + 1]
```

Responda:

(a) Prove que o predicado $P(t)$ = *"Antes da t-ésima iteração começar, temos que $j = t$ e $A[j]$ é maior do que os elementos de $A[1..j-1]$."* é uma invariante do laço da linha 3.

(b) Prove que o predicado $R(\ell)$ = "Antes da ℓ-ésima iteração começar, temos que $i = n + 1 - \ell$, o subvetor $A[i + 1..n]$ está ordenado e contém os maiores elementos de A." é uma invariante do laço externo.

(c) Demonstre a correção do *Bubble Sort*.

2. Seja $A[1..n]$ um vetor com n elementos inteiros positivos — isto é, para qualquer $A[i]$, com $1 \leq i \leq n$, $A[i] > 0$. O vetor não necessariamente está ordenado.

(a) Escreva um pseudo-código que implemente a seguinte estratégia para ordenar os elementos de A em ordem crescente:

> *Primeiro, encontre o menor elemento de A e troque-o com o elemento na posição $A[1]$. Após isso, encontre o segundo menor elemento de A e troque-o com o elemento na posição $A[2]$. Repita o processo para os primeiros $n - 1$ elementos de A. Ao final de todas as iterações, o vetor A deve estar ordenado de maneira crescente.*

O problema de ordenação consiste em modificar as posições dos elementos do vetor A de forma que $A[i] \leq A[j], \forall i, j \in \{1, \ldots, n\}, i \leq j$.

Dica: repare que são necessários 2 laços aninhados para que seja possível fazer a implementação: um laço externo para realizar o procedimento descrito, e um laço interno para encontrar, a cada iteração, o respectivo menor elemento de A.

(b) Com base no seu pseudo-código, prove que os seguintes itens compõem uma invariante (do laço mais externo):

P(t) = "antes da t-ésima iteração começar, $A[1..i-1]$ está em ordem crescente e todos os valores de $A[i..n]$ são maiores ou iguais aos elementos de $A[1..t-1]$";

(c) É necessário percorrer todos os elementos de A durante a ordenação, ou o algoritmo proposto funciona corretamente percorrendo os elementos de $A[1..n-1]$? Prove seu argumento.

3. Considere o seguinte pseudo-código, que recebe um número inteiro positivo n e retorna *Verdadeiro* caso n seja uma potência de 2 e *Falso* caso contrário:

```
1: Função POTÊNCIA-DE-DOIS(n)
2:     a ← n
3:     Enquanto a > 1 e a ≡ 0 (mod 2) faça
4:         a ← a/2
5:     Se a = 1 então
6:         Devolve Verdadeiro
7:     Devolve Falso
```

(a) O laço termina em algum momento? Por que?

(b) Prove que a seguinte afirmação é uma invariante de laço para o laço da linha 3.

$\mathrm{P}(t) =$ "antes da t-ésima interação começar, $a = \frac{n}{2^{t-1}}$".

(c) Usando a invariante de laço obtida no exercício anterior, prove a correção do algoritmo POTÊNCIA-DE-DOIS.

4. Termine a demonstração da invariante da busca binária (veja as notas de aula).

5. Seja $\mathbb{N}$ o conjunto de números naturais. O fatorial de um número $n \in \mathbb{N} \cup \{0\}$, denotado por $n!$, é definido como

$$n! = n \times (n-1) \times (n-1) \times \cdots \times 2 \times 1.$$

Considere o seguinte pseudo-código que calcula o fatorial de um número $n \in \mathbb{N}$:

```
Função FATORIAL(n)
    fatorial ← 1
    Para i = 1 até n faça
        fatorial ← fatorial ×i
    Devolve fatorial
```

(a) Escreva uma invariante de laço para o pseudo-código acima.

(b) Prove que sua invariante de laço está correta.