PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
DEPARTAMENTO DE INFORMÁTICA

AULA 9 - Argumentação de Corretude Continuação

Iniciais dos alunos: *AVC, JPP, PC*

Disciplina: *Programação Modular (INF1301)* – Professor: *Flavio Bevilacqua*
:

Algoritmo 1: Insertion Sort

AE $\longrightarrow$
INÍCIO
IND $\longleftarrow$ 1
ENQUANTO *IND* $\leqslant$ *LIMITE-LÓGICO* **FAÇA**
ATUAL $\longleftarrow$ ELE[IND]
AUX $\longleftarrow$ IND - 1
ENQUANTO *AUX* $>$ *0 E ELE[AUX]* $>$ *ATUAL* **FAÇA**
ELE[AUX + 1] $\longleftarrow$ ELE[AUX]
AUX $\longleftarrow$ AUX - 1
FIM ENQUANTO
ELE[AUX + 1] $\longleftarrow$ ATUAL
IND $\longleftarrow$ IND + 1
FIM ENQUANTO
FIM
AS $\longrightarrow$

Argumentação de Sequência 1

AE: Existe um vetor a ser ordenado.

AS: Vetor está vazio ou foi ordenado.

AI 1: IND aponta para o primeiro elemento do vetor.

Argumentação de Repetição 1

AE: AI 1.

AS: AS geral.

AINV:

- Existem dois conjuntos: a ordenar e já ordenados.

- IND aponta para o elemento a ordenar.

(1) AE $\Longrightarrow$ AINV

- Pela AE, IND aponta para o primeiro elemento do vetor. Todos os elementos estão no conjunto a ordenar e o conjunto já ordenados está vazio. Logo, vale a AINV.

(2) AE && (Condição == False) $\Longrightarrow$ AS

- Pela AE, IND == 1. Para que (Condição == False), LIMITE-LÓGICO = 0, ou seja, vetor está vazio. Neste caso, vale a AS.

(3) AE && (Condição == True) $\oplus$ B $\Longrightarrow$ AINV

- Pela AE, IND aponta para o primeiro elemento do vetor que não está vazio. Este elemento já se encontra no local ordenado. Com isso, os dois conjuntos existem e um elemento de a ordenar passou para já ordenados. Logo, vale a AINV.

(4) AINV && (Condição == True) $\oplus$ B $\Longrightarrow$ AINV

- Para que a AINV continue valendo, B deve garantir que um elemento passe do conjunto a ordenar para já ordenado e IND seja reposicionado.

(5) AINV && (Condição == False) $\oplus$ B $\Longrightarrow$ AS

- Se (Condição == False), IND ultrapassou o LIMITE-LÓGICO, ou seja, todos os elementos passaram para o conjunto já ordenados. Como o vetor está ordenado, vale a AS.

(6) Término

- Como a cada ciclo, um elemento é retirado do conjunto a ordenar, e este conjunto possui um número finito de elementos, a repetição terminará em um número finito de passos.

Argumentação de Sequência 2

AE (seq2) = AS (seq2) = AINV

AI 2: ATUAL recebe elemento a ser reposicionado.

AI 3: AUX aponta para o último elemento do conjunto já ordenado (caso não esteja vazio) ou AUX $\longleftarrow$ 0.

AI 4: Local para onde ATUAL será reposicionado foi definido.

AI 5: ATUAL foi reposicionado.

OBS: o último bloco (IND $\longleftarrow$ IND + 1) garante que IND foi reposionado para o próximo elemento a ordenar ou ultrapassou o LIMITE-LÓGICO. Vale a AINV ou a AS geral se o vetor está ordenado.

Argumentação de Repetição 2

AE: AI 3.

AS: AI 4.

AINV:

- Existem dois conjuntos: maiores e possíveis menores.
- AUX aponta para elemento de possíveis menores.

1. AE $\Longrightarrow$ AINV
   - Pela AE, AUX aponta para o último elemento de possíveis menores e o conjunto de maiores está vazio. Vale a AINV.
2. AE && (Condição == False) $\Longrightarrow$ AS
   - Para que (Condição == False) antes do primeiro ciclo da repetição, AUX == 0 e IND == 1. IND aponta para o primeiro elemento a ser reposicionado e ele já se encontra posicionado, valendo a AS.
3. AE && (Condição == True) $\oplus$ B $\Longrightarrow$ AINV

- Como o primeiro ciclo executou, o elemento apontado por AUX é maior do que ATUAL. Com isso, ele passou do conjunto de possíveis menores para maiores e AUX foi reposicionado. Vale a AINV.

(4) AINV && (Condição == True) $\oplus$ B $\Longrightarrow$ AINV

- Para que a AINV continue valendo, B deve garantir que um elemento passe do conjunto a possíveis menores para maiores e AUX seja reposicionado.

(5) AINV && (Condição == False) $\oplus$ B $\Longrightarrow$ AS

- Se (Condição == False), AUX == 0, ou seja, todos os elementos de possíveis foram reposicionados no conjunto maiores. Isso significa que ATUAL é menor do que todos. Logo, a posição em que ele será reposicionado é AUX == 0 + 1 == 1. Vale a AS.

- Se (Condição == False), ELE[AUX] $\longleftarrow$ ATUAL, se ocorrer este teste, todos os elementos à esquerda de AUX serão menores que ATUAL. Com isso, foi definido o local em que ATUAL será reposicionado, valendo a AS.

(6) Término

- Como o número de elementos do conjunto possíveis menores é finito e cada ciclo retira um elemento deste conjunto, a repetição terminará em um número finito de passos.