

Programação Modular: Lista 2

Entrega no dia 16 de outubro de 2019 às 17h

Professor Flavio Bevilacqua

Antônio Vasconcellos Chaves

Engenharia da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
antoniovasconcelloschaves@gmail.com

João Pedro Paiva

Ciência da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
joaopedrordepaiva@gmail.com

Pedro Moreira Costa

Engenharia da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
pedromoreiramcosta@gmail.com

Questão 1

Apresente a estrutura de decomposição sucessiva do algoritmo de quicksort apontando um componente concreto, um componente abstrato e um conjunto solução.

Solução

Questão 2

Faça a argumentação de corretude completa de uma pesquisa binária em um vetor.

Solução

Algoritmo 1: Busca Binária em Vetor

```

AE →
  INÍCIO
    COMEÇO ← 1
    FINAL ← LIMITE-LÓGICO
    ENQUANTO  $COMEÇO \leq FINAL$  FAÇA
      ATUAL ←  $(COMEÇO + FINAL) / 2$ 
      SE  $VETOR[ATUAL] == PARÂMETRO-BUSCADO$  ENTÃO RETORNA ATUAL
      SE  $VETOR[ATUAL] < PARÂMETRO-BUSCADO$  ENTÃO COMEÇO ← ATUAL + 1
      SENÃO FINAL ← ATUAL - 1
    FIM ENQUANTO
    RETORNA -1
  FIM
AS →

```

Argumentação de Sequência 1

AE: Existe um número a ser buscado em um vetor ordenado.

AS: PARÂMETRO-BUSCADO está na posição retornada, ou não está no vetor e o valor retornado é -1.

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

AI 2: FINAL aponta para o LIMITE-LÓGICO do vetor.

AI 3: PARÂMETRO-BUSCADO não está no vetor.

Argumentação de Repetição 1

AE: AI 2.

AS: PARÂMETRO-BUSCADO está na posição retornada, ou não está no vetor.

AINV:

- Existem dois conjuntos: pode conter PARÂMETRO-BUSCADO e não contém PARÂMETRO-BUSCADO.
- COMEÇO e FINAL apontam para os limites inferior e superior, respectivamente, do conjunto pode conter PARÂMETRO-BUSCADO.

① $AE \implies AINV$

- Pela AE, FINAL aponta para o LIMITE-LÓGICO do vetor. Todos os elementos estão no conjunto pode conter PARÂMETRO-BUSCADO e o conjunto não contém PARÂMETRO-BUSCADO está vazio. Logo, vale a AINV.

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

- Pela AE, $FINAL == LIMITE-LÓGICO$. Para que $(Condição == False)$, $LIMITE-LÓGICO == 0$, ou seja, vetor está vazio. Neste caso, vale a AS, já que o $PARÂMETRO-BUSCADO$ não está no vetor.

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

- Pela AE, $FINAL$ aponta para o último elemento do vetor que não está vazio. Metade dos elementos do conjunto pode conter $PARÂMETRO-BUSCADO$ passarão para o conjunto não contém $PARÂMETRO-BUSCADO$, e $COMEÇO$ e $FINAL$ serão reposicionados apontando para os limites do novo conjunto pode conter $PARÂMETRO-BUSCADO$. Com isso, os dois conjuntos existem e $COMEÇO$ e $FINAL$ apontam para os limites do conjunto pode conter $PARÂMETRO-BUSCADO$. Logo, vale a AINV.

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

- Para que a AINV continue valendo, B deve garantir que metade dos elementos do conjunto pode conter $PARÂMETRO-BUSCADO$ passem para o conjunto não contém $PARÂMETRO-BUSCADO$, e $COMEÇO$ e $FINAL$ sejam reposicionados apontando para os limites do novo conjunto pode conter $PARÂMETRO-BUSCADO$.

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

- Se $(Condição == False)$, o limite inferior superou o limite superior do conjunto pode conter $PARÂMETRO-BUSCADO$, ou seja, todos os elementos passaram para o conjunto não contém $PARÂMETRO-BUSCADO$ e o conjunto pode conter $PARÂMETRO-BUSCADO$ não existe mais. Como o $PARÂMETRO-BUSCADO$ não está no vetor, vale a AS.

⑥ **Término**

- Como a cada ciclo, metade dos elementos do conjunto pode conter $PARÂMETRO-BUSCADO$ são retirados, 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 4: $ATUAL$ aponta para o meio do conjunto pode conter $PARÂMETRO-BUSCADO$.

Argumentação de Seleção 1

AE: AI 4.

AS: AINV ou AS geral.

① $AE \ \&\& \ (Condição == True) \oplus B1 \implies AS$

Pela AE, $ATUAL$ aponta para o meio do conjunto pode conter $PARÂMETRO-BUSCADO$. Como $(Condição == True)$, $ATUAL$ aponta para o $PARÂMETRO-BUSCADO$. Neste caso, executa B1 que retorna a posição de $PARÂMETRO-BUSCADO$, valendo a AS.

② $AE \ \&\& \ (Condição == False) \oplus B2 \implies AS$

Pela AE, $ATUAL$ aponta para o meio do conjunto pode conter $PARÂMETRO-BUSCADO$. Como $(Condição == False)$, $ATUAL$ não aponta para o $PARÂMETRO-BUSCADO$. Neste caso, executa B2 que passa metade dos elementos do conjunto pode conter $PARÂMETRO-BUSCADO$ para o conjunto não contém $PARÂMETRO-BUSCADO$. Vale a AS pois $COMEÇO$ e $FINAL$ apontam para os limites inferior e superior, respectivamente, do conjunto pode conter $PARÂMETRO-BUSCADO$.

Argumentação de Seleção 2

AE (sel2) = AE (sel1) e ATUAL não aponta para o PARÂMETRO-BUSCADO.

AS: AINV e metade inferior ou superior dos elementos do conjunto pode conter PARÂMETRO-BUSCADO foi passada para o conjunto não contém PARÂMETRO-BUSCADO.

① AE && (Condição == True) \bigoplus B1 \implies AS

Pela AE, ATUAL aponta para o meio do conjunto pode conter PARÂMETRO-BUSCADO mas não aponta para o PARÂMETRO-BUSCADO. Como (Condição == True), o elemento apontado por ATUAL é menor do que PARÂMETRO-BUSCADO. Neste caso, executa B1 que redefine COMEÇO, passando a metade inferior dos elementos do conjunto pode conter PARÂMETRO-BUSCADO para o conjunto não contém PARÂMETRO-BUSCADO e valendo a AS.

② AE && (Condição == False) \bigoplus B2 \implies AS

Pela AE, ATUAL aponta para o meio do conjunto pode conter PARÂMETRO-BUSCADO mas não aponta para o PARÂMETRO-BUSCADO. Como (Condição == False), o elemento apontado por ATUAL é maior ou igual que PARÂMETRO-BUSCADO. Neste caso, executa B2 que redefine FINAL, passando a metade superior dos elementos do conjunto pode conter PARÂMETRO-BUSCADO para o conjunto não contém PARÂMETRO-BUSCADO e valendo a AS.