

Projeto e Análise de Algoritmos (2022-1)

Primeira avaliação (Gabarito)

Prof. Flávio L. C. de Moura

02 de agosto de 2022

1. O pseudocódigo para o algoritmo recursivo de busca binária em um vetor ordenado de inteiros com n elementos é dado a seguir:

```
1 if  $high < low$  then
2   | return -1;
3 end
4  $mid = \lfloor (high + low)/2 \rfloor$ ;
5 if  $key > A[mid]$  then
6   | return BinarySearch( $A, mid + 1, high, key$ );
7 end
8 else
9   | if  $key < A[mid]$  then
10  |   | return BinarySearch( $A, low, mid - 1, key$ );
11  |   end
12  |   else
13  |   | return  $mid$ ;
14  |   end
15 end
```

Algorithm 1: BinarySearch($A[1..n], low, high, key$)

- (a) (2.0 ponto) Faça a análise da complexidade do melhor caso para este algoritmo.

solução

- (b) (2.5 pontos) Faça a análise da complexidade do pior caso para este algoritmo.

solução

- (c) (3.0 pontos) A correção deste algoritmo pode ser estabelecida em duas etapas. A primeira dela consiste em provar que se a chave key não ocorre no vetor $A[1..n]$, então BinarySearch($A[1..n], 1, n, key$) retorna o valor -1. Prove o lema a seguir:

Seja $A[1..n]$ um vetor ordenado de inteiros distintos. Mostre que se a chave key não ocorre em $A[1..n]$, então BinarySearch($A[1..n], 1, n, key$) retorna o valor -1.

Dica: Indução (forte) sobre o tamanho n do vetor.

solução

2. (2.5 pontos) Sejam $f(n)$, $g(n)$ e $h(n)$ funções não-negativas tais que $f(n) = O(h(n))$ e $g(n) = O(h(n))$. Prove, utilizando as definições de notação assintótica, que $f(n) + g(n) = O(h(n))$.

solução