Tópicos Avançados em Estrutura de Dados

Atividade 8

Bruna Galastri Guedes	18.00189-0
Daniel Ughini Xavier	18.00022-3
Rodolfo Cochi Bezerra	18.00202-0
Vítor Martin Simoni	18.00050-9
Leonardo Cury Haddad	18.00442-3
Leonardo de Barros Rodrigues	18.02401-7

17/05/2020

Questão 1

Segue o código do algoritmo de Busca Binária, versão Recursiva, na linguagem C:

```
#include<stdio.h>
int BinariaRecursiva(int A[], int inicio, int fim, int elemento) {
   if(inicio > fim) return -1;
      int mid = (inicio+fim)/2;
   if( A[mid] == elemento ) return mid;
   else if( elemento < A[mid] )
      BinariaRecursiva(A, inicio, mid-1, elemento);
   else
      BinariaRecursiva(A, mid+1, fim, elemento);
}
```

a) A complexidade do algoritmo é O(log n).

b) A justificativa para tal complexidade é a própria recorrência da busca binária:

$$\begin{aligned} T(n) &= T(\tfrac{n}{2}) + 1 \\ &= (T(\tfrac{n}{4}) + 1) + 1 \\ &= ((T(\tfrac{n}{8}) + 1) + 1) + 1 \\ &= T(\tfrac{n}{2^3}) + 3 \\ &\cdots \\ &= T(\tfrac{n}{2^k}) + k \end{aligned}$$

Considerando o caso onde n = 2^k, e aplicando log na base 2 nos dois lados, temos:

$$\begin{aligned} log_2 n &= log_2(2^k) \\ log_2 n &= k * log_2 2 \\ k &= log_2 n \end{aligned}$$

Substituindo o caso hipotético na função encontrada anteriormente, temos:

$$\begin{aligned} T(n) &= T(1) + k \\ T(n) &= 1 + k \end{aligned}$$

Substituindo o k:

$$T(n) = T(1) + log_2 n$$

Sendo assim, a complexidade do algoritmo recursivo da busca binária é O(log n).