

Relações de Recorrência: Introdução

Análise de Algoritmos – Ciência da Computação



Prof. Daniel Saad Nogueira
Nunes

IFB – Instituto Federal de Brasília,
Campus Taguatinga



Introdução

Recorrência

Em algoritmos recursivos, o tempo de execução pode ser descrito por recorrências, que são equações ou desigualdades que descrevem uma função em termos de seu valor em entradas menores.

Exemplo

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ 2T(n/2) + \Theta(n), & n > 1 \end{cases}$$



Introdução

Exemplo: Fibonacci

Algorithm 1: FIB(n)

Input: $n \in \mathbb{N} \cup \{0\}$

Output: i -ésimo número da sequência de Fibonacci

```
1 if(  $n \leq 2$  )
2   | return 1
3 return FIB( $n - 1$ ) + FIB( $n - 2$ )
```



Introdução

Exemplo: Fibonacci

A relação de recorrência é dada por:

$$T(n) = \begin{cases} \Theta(1), & n \leq 2 \\ T(n-1) + T(n-2) + \Theta(1), & n > 2 \end{cases}$$



Introdução

Exemplo: Busca Binária

Algorithm 2: BSEARCH(V, l, r, k)

Input: $V[0, n - 1], 0 \leq l, r < n, k$

Output: $x, V[x] = k$, ou -1 caso $V[x] \neq k, 0 \leq x < n$

```
1 if(  $l > r$  )
2   | return  $-1$ 
3  $m \leftarrow l + \lfloor \frac{r-l}{2} \rfloor$ 
4 if(  $k = V[m]$  )
5   | return  $m$ 
6 else if(  $k < V[m]$  )
7   | return BSEARCH( $V, l, m - 1, k$ )
8 return BSEARCH( $V, m + 1, r, k$ )
```



Introdução

Exemplo: Busca Binária

A relação de recorrência é dada por:

$$T(n) = \begin{cases} \Theta(1), & n = 0 \\ T(\lfloor \frac{n}{2} \rfloor) + \Theta(1), & n > 0 \end{cases}$$



Introdução

- Como analisar algoritmos recursivos?
- Precisamos identificar cotas superiores e inferiores a partir da relação de recorrência.



Introdução

Recorrências

Para encontrar a complexidade de um algoritmo descrito por uma recorrência, geralmente recorre-se aos seguintes métodos:

- 1 Método da Substituição (Indução);
- 2 Método da Iteração;
- 3 Método Master;