INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Matemática Discreta

Adriana Padua Lovatte

Demonstrações, Recorrências e Análise de Algoritmos

- Técnicas de Demonstração
- Indução
- Mais Sobre Demonstração de Correção
- Recursividade e relações de Recorrência
- Análise de Algoritmo

Mais Sobre Demonstração de Correção

Definição: Uma definição onde o item sendo definido aparece como parte da definição é chamada de uma **definição recorrente.**

Definição por recorrência.

1) Condição básica
2) Passo de indução

Recursividade e Relações de Recorrência

Ex:

$S(1) = 2$

$S(n) = 2S(n - 1), n > 1$

Ex:

$T(1) = 1$

$T(n) = T(n - 1) + 3, n > 1$

Ex: Sequência de Fibonacci

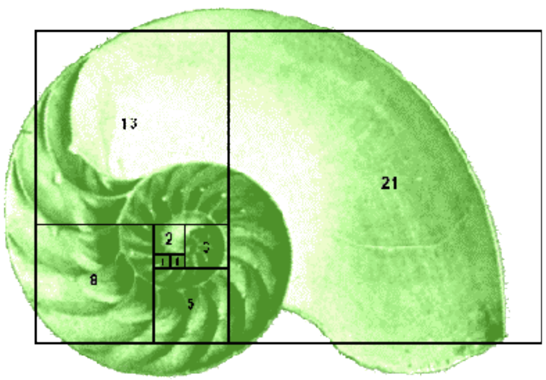
$F(1) = 1$

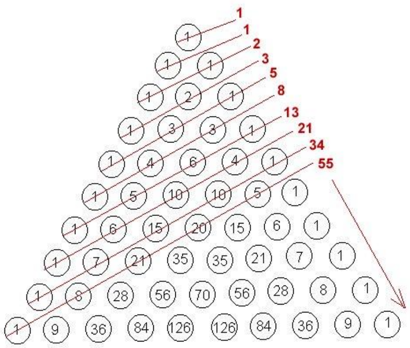
$F(2) = 1$

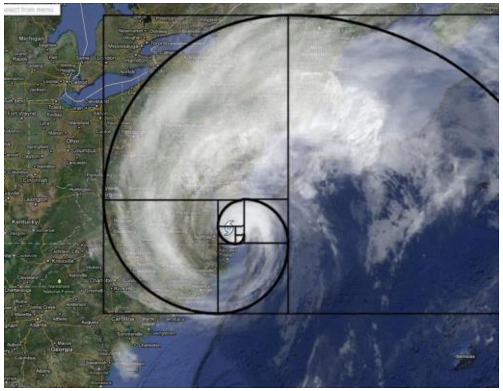
$F(n) = F(n - 2) + F(n - 1), n > 2$

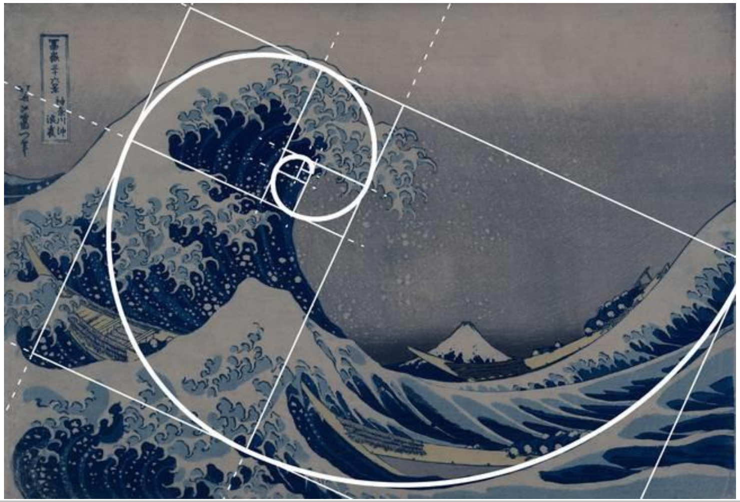
Recursividade e Relações de Recorrência

Ex: Sequência de Fibonacci









Recursividade e Relações de Recorrência

Ex: Sequência de Fibonacci

$F(1) = 1$

$F(2) = 1$

$F(n) = F(n - 2) + F(n - 1), n > 2$

Ex:Prove que na sequência de Fibonacci

$F(n + 4) = 3F(n + 2) - F(n)$, para $n \geq 1$.

Por indução

Por demonstração direta

Recursividade e Relações de Recorrência

Operações definidas por recorrência

Exemplo: Definição recorrente de exponenciação

$a^0 = 1$

$a^n = a^{(n-1)}a$; para $n \geq 1,\ n \in Z$

Exemplo: Definição recorrente para multiplicação de dois inteiros positivos *m* e *n*.

$m(1) = m$

$m(n) = m(n-1) + m$; para $n \geq 2$

Recursividade e Relações de Recorrência

Defina recursivamente $n!$

$0! = 1$ *(Base)*

$n! = n.(n-1)!$

Conjuntos definidos por recorrência

Um conjunto diferencia de uma sequência na ordem, isto é , na sequência os elementos tem uma ordem, enquanto em um conjunto os elementos não precisam ter uma ordem.

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Recursividade e Relações de Recorrência

Conjuntos definidos por recorrência

Exemplo: Conjunto das fórmulas bem formuladas.

1 – Qualquer letra de proposição é uma fbf (Base)

2 – Se P e Q são fbfs, então $(P \wedge Q)$; $(P \vee Q)$; $(P \rightarrow Q)$; $\sim P$ *e* $(P \leftrightarrow Q)$ *também o são* (Passo de indução ou recorrência)

Por exemplo Se A,B e C são fbfs pela regra 1, pela regra 2 $((A \wedge B) \rightarrow C)$ é uma fbf

Recursividade e Relações de Recorrência

Na sequência definida por:

$S(1) = 2$

$S(n) = 2S(n – 1), n > 1$

Podemos escrever uma função usando uma abordagem iterativa de laço ou um função que use diretamente a definição recorrente de S.

Recursividade e Relações de Recorrência

Ex: **Iterativamente**

Função S(n)
Inteiro i
ValorCorrente
se n=1 **então**
    retorne 2
senão
    i=2
    ValorCorrente = 2
    enquanto i <= n **faça**
        ValorCorrente = 2* ValorCorrente
        i = i + 1
    fim do enquanto
    retorna ValorCorrente
fim do senão
fim de S

Usando Recorrência

Função S(n)
se n=1 **então**
    retorne 2
senão
    retorne 2*S(n-1)
fim do senão
fim de S

Recursividade e Relações de Recorrência

Exemplo: Escreva uma função recorrente para a seguinte operação.

$m(1) = m$
$m(n) = m(n\text{-}1) + m$; para $n \geq 2$

Produto(m,n)
se $n = 1$ **então**
    retorne m
senão
    retorne Produto (m,n-1) + m
fim do senão
fim da função Produto

OBS: Note que deve haver uma redução do problema a ser tratado

Exemplo: Escreva uma função iterativa e uma função recorrente para a sequência de Fibonacci.

Na função a seguir vamos considerar a sequência de Fibonacci dada por :

$F(1) = 1$

$F(2) = 1$

$F(n) = F(n-2) + F(n-1), n > 2$

Exemplo: Escreva uma função iterativa e uma função recorrente para a sequência de Fibonacci.

```
função Fibonacci (n)
Inteiro: ant1, ant2: 1;              aux, i
se ( n=1 ou n=2 )então
        retorne 1
senão
       i=3
       aux = 0
       enquanto i <= n faça
                          i = i + 1;
                          aux= ant2;
                          ant2 = ant2+ant1;
                           ant1 = aux;
        fim do enquanto
    Retorna ant2
     fim do senão
fim de S
```

Forma
Iterativa

Exemplo: Escreva uma função iterativa e uma função recorrente para a sequência de Fibonacci.

Fibonacci (n)
se (n = 1 ou n=2)**então**
  retorne *1*
senão
  retorne Fibonacci (n-1)+Fibonacci (n-2)
fim do senão
fim da função Fibonacci

Forma Recorrente

Recursividade e Relações de Recorrência

Exemplo: Escreva uma função recorrente para a seguinte sequência.

$T(1) = 1$

$T(n) = T(n - 1) + 3, n > 1$

Recursividade e Relações de Recorrência

Ex: Escreva uma função recorrente para a seguinte sequência.

$T(1) = 1$

$T(n) = T(n - 1) + 3, n > 1$

Recursividade e Relações de Recorrência

Ex: **Ordenação de dados**

OrdenaçãoPorSeleção(lista L, int n)
se $n = 1$ **então**
  a ordenação está completa, escrever a lista
senão
  encontre o índice i do maior item em lista entre 1 e n
  permute $L[i]$ e $L[n]$
  OrdenaçãoPorSeleção(L, $n - 1$)
fim do senão
fim da função OrdenaçãoPorSeleção

http://pt.wikipedia.org/wiki/Ficheiro:Selection-Sort-Animation.gif

Algoritmo de Busca Binária

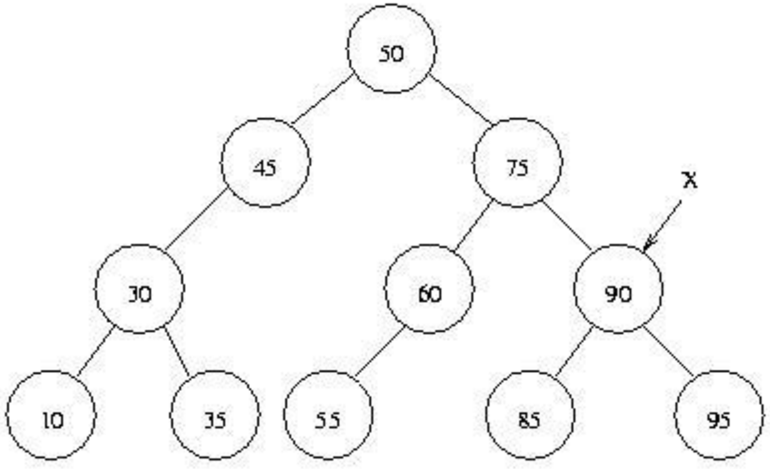
- Lista deve estar ordenada ;
- Objetivo: Encontrar um elemento X nesta lista;
- Considere o elemento K do meio da lista;
  - Se X=K encontrou ;
    - Se X<K procure apenas na primeira metade (esquerda);
    - Se X>K procure apenas na segunda metade (direita);
- Repita o processo até que o X seja encontrado ou que a sublista em questão esteja vazia

http://pt.wikipedia.org/wiki/Ficheiro:Selection-Sort-Animation.gif

ALGORÍTMO BUSCA BINÁRIA

BuscaBinária(lista L, int j, int k item x)
Inteiro: i
i=1;
se $i > j$ **então**
    não encontrado
Senão
    encontre o índice k do item do meio na lista *(i + j)/2*
    se x = item do meio **então**
        encontrado
    senão
        se x < item do meio **então**
            BuscaBinária(L, i, k-1, x)
        senão
            BuscaBinária(L, k+1, j, x)
        fim do senão
    fim do senão
fim do senão
fim da função BuscaBinária

Ilustração



INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Resolvendo Relações de Recorrência

Expandir, Conjecturar e Verificar (Indução matemática)

Exemplo: Dada a recorrência

$S(1) = 2$

$S(n) = 2S(n - 1), n > 1.$

Encontre uma **forma fechada** para $S(n)$.

Dicas:

1) Aplique a recorrência para n, n-1, n-2, n-3
2) Note que a recorrência termina com ...

Após a conjectura a verificação pode ser feita por indução

Exemplo:

$T(1) = 1$

$T(n) = T(n - 1) + 3, n > 1.$

Encontre uma **formula fechada** para $T(n)$.

Resolvendo Relações de Recorrência

Expandir, Conjectursar e Verificar (Indução matemática)

Exemplo:

$T(1) = 1$

$T(n) = 2.T(n - 1) + 1, n > 1.$

Encontre uma **formula fechada** para $T(n)$.

Usar fórmula de soma de uma PG finita. $S_n = \frac{a_1(q^n - 1)}{q - 1}$

Recursividade e Relações de Recorrência

Resolvendo Relações de Recorrência

Tipo de Recorrências

Linear:

S(n) é uma relação de recorrência linear se os valores anteriores de S que aparecem na definição tem potência 1.

$S(n) = f_1(n)S(n-1) + f_2(n)S(n-2) + \cdots + f_k(n)S(n-k) + g(n)$, *onde os* f_i e g podem ser expressões que envolvem n

→

Recursividade e Relações de Recorrência

Tipo de Recorrências

Primeira ordem: ocorre quando o n-ésimo termo depende apenas do termo $n - 1$.

$$S(n) = cS(n - 1) + g(n)$$

OBS: Se $g(n) = 0$ a recorrência é dita **homogênea**

Forma fechada para uma recorrência de primeira ordem não homogênea.

$$S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i} g(i)$$

Recursividade e Relações de Recorrência

Resolvendo Relações de Recorrência

Tipo de Recorrências

Primeira ordem: ocorre quando o n-ésimo termo depende apenas do termo $n - 1$.

$$S(n) = cS(n - 1) + g(n)$$

OBS: Se $g(n) = 0$ a recorrência é dita **homogênea**

Fórmula da Solução: $$S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i} g(i)$$

Exemplo:

$S(1) = 2$

$S(n) = 2S(n - 1), n > 1.$

→ $$S(n) = 2^{n-1}S(1) + \sum_{i=2}^{n} 2^{n-i} * 0 = 2^n$$

Recursividade e Relações de Recorrência

Resolvendo Relações de Recorrência

Resumo

Método	Passos
Expandir, Conjecturar, verificar	1. Use a relação de recorrência repetidamente até poder adivinhar o padrão. 2. Decida qual será o padrão quando $n - k = 1$. 3. Verifique a fórmula resultante por indução.
Fórmula da Solução	1. Coloque sua relação de recorrência da forma $S(n) = cS(n-1) + g(n)$ para encontrar c e $g(n)$. 2. Use c, $g(n)$ e $S(1)$ na Formula. $S(n) = c^{n-1}S(1) + \sum_{i=2}^{n} c^{n-i} g(i)$ 1. Calcule o somatória para obter a expressão final.

Recursividade e Relações de Recorrência

Resolvendo Relações de Recorrência

Ex: Encontre uma solução em forma fechada para a relação de recorrência
$S(1) = 4$
$S(n) = 2S(n-1) + 3$, para $n \geq 2$

1ª opção: Usando a formula temos $c = 2$, $g(n) = 2$ e $S(1) = 4$, logo

$$S(n) = 2^{n-1}(4) + \sum_{i=2}^{n} 2^{n-i}(3)$$

Resolver o somatório.

2ª opção: Expandir, Conjecturar, verificar

Recursividade e Relações de Recorrência

Resolvendo Relações de Recorrência

Ex: Encontre uma solução em forma fechada para a relação de recorrência
$T(1) = 2$
$T(n) = T(n - 1) + (n+1)$, para $n \geq 2$

Usando a formula temos $c = 1$, $g(n) = n+1$ e $S(1) = 2$, logo

$$S(n) = 1^{n-1}(2) + \sum_{i=2}^{n} 1^{n-i}(i+1)$$

Lista Mínima de Exercícios

Seção 2.4: 4, 7, 10, 11, 14, 15, 16, 20, 21, 25, 29, 33, 34, 38, 50, 51, 58, 59, 61, 62, 63, 64, 65, 66, 73, 76, 78, 79, 80, 88 e 89.