INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Matemática Discreta

Adriana Padua Lovatte

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Demonstrações, Recorrências e Análise de Algoritmos

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

Análise de Algoritmo

Objetivo: Comparar dois algoritmos para escolher o **melhor**.

O que é MELHOR?

O mais fácil de implementar?

O mais eficiente? (isto é, realiza a mesma tarefa com um número menor de operações).

Análise de Algoritmo

BuscaSequencial(lista L, int n, item x)
inteiro i
$i = 1$
enquanto $L[i] \neq x$ e $i < n$ **faça**
    $i = i + 1$
fim do enquanto
se $L[i] = x$ **então**
    Encontrado
senão
    Não Encontrado
fim do senão
fim da BuscaSequencial

<u>Número de operações:</u>

- Melhor caso
- Caso médio
- Pior caso

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Análise de Algoritmo

para *i* = *1* até *n* **faça**
menor = aluno[i].teste[1]
maior = aluno[i].teste[1]
  para *j* = *2* até *m* **faça**
    soma = soma + aluno[i].teste[j]
    se *aluno[i].teste[j] < menor* **então**
      menor = aluno[i].teste[j]
    fim do se
  fim do para
soma = soma – menor
escreva ("nota do aluno", *i*, "é", *soma*)
fim do para

<u>O que esse algoritmo faz?</u>

<u>Número de operações:</u>

- Melhor caso?
- Caso médio?
- Pior caso?

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Análise de Algoritmo

Análise Usando Relações de Recorrência

Quantas **comparações** são necessárias, no pior caso, para o algoritmo de busca binária?

Análise de Algoritmo

Análise Usando Relações de Recorrência

Quantas comparações são necessárias, no pior caso, para o algoritmo de busca binária?

Note que fazemos uma comparação e se não for encontrado, a mesma busca recomeça com metade da lista!

Logo, temos

$$C(n) = 1 + C\left(\frac{n}{2}\right)$$

Análise de Algoritmo

Análise Usando Relações de Recorrência

Logo para análise do número de comparações de uma busca binária basta resolver a recorrência.

$$C(n) = 1 + C\left(\frac{n}{2}\right)$$

, para $n \geq 2$. Adicionalmente, podemos supor $n = 2^m$

e a condição básica $C(1) = 1$

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Análise de Algoritmo

Análise Usando Relações de Recorrência

Estratégia famosa: **Dividir para Conquistar**.

Forma Geral: $S(n) = cS\left(\frac{n}{2}\right) + g(n)$, para $n \geq 2$, $n = 2^m$

Tarefa: Resolver a recorrência acima.

Análise de Algoritmo

Análise Usando Relações de Recorrência

Estratégia famosa: **Dividir para Conquistar**.

Forma Geral: $S(n) = cS\left(\frac{n}{2}\right) + g(n)$, para $n \geq 2$, $n = 2^m$

Tarefa: Resolver a recorrência acima.

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

Análise de Algoritmo

Análise Usando Relações de Recorrência

Ex: Resolva as relações de recorrências.

$T(1) = 1$

$$T(n) = 1 + T\left(\frac{n}{2}\right) \qquad \text{, supor } n \geq 2,\ n = 2^m$$

Lembrar:

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

$T(1) = 3$

$$T(n) = 2T\left(\frac{n}{2}\right) + 2n$$

Análise de Algoritmo

Análise Usando Relações de Recorrência

Lembra do algoritmo de Euclides para encontrar o MDC?

Vamos encontrar o número de divisões realizadas por tal algoritmo.

MDC(a,b) /* exercício 66 da Seção 2.4 */
calcule $a = qb + r$, para $0 \leq r < b$
se $r = 0$ **então**
    Retorna b
senão
    MDC(b,r)
fim senão
fim MDC

Cota superior: valor superior para o número de operações efetuadas.

Lista Mínima de Exercícios

Seção 2.5: 2, 4, 6, 7, 9, 10, 16, 17, 20, 21, 22.