-- Inteiros Grandes

-Soma

seja um processador com tamanho de palavra p (32 ou 64 bits)

e k o numero de bits de um numero da operacao de soma ou subtracao

temos a complexidade como O(k/p);

logo a complexidade seria O(k)

k eh a representacao binaria de um numero n

entao O(K) = O(log(n))

-Multiplicacao

Sendo k o número de bits de x e y,

qual a complexidade de tempo para este algoritmo?

São feitas k somas de complexidade O(k), logo:

k \* O(k) = O(k2)

ou O(log2 n)

T(k) = 3\*T(k/2) + O(k)

T(1) = 1

(pelo teorema mestre cai no caso 3 e nao da pra resolver pois

a\*f(k/b) <= c\*f(k) -> falso )

fazendo recorencia cheguei em

3^log(k) + ∑ 3^i O(K/2^i) i = 0 ateh i = log(k)

-- RSA

Baseia na ideia que eh facil encontrar um numero primo

porem eh dificil fatorar o produto de dois primos

No sistema de cripto RSA cada participante tem chave publica e privada

Como funciona RSA :

Bob obtem a chave publica de Alice

Bob usa a chave para codificar uma msg e envia para Alice

alice usa sua chave privada para recuperar a msg

chave publica na forma(e,n)

n = p\*q onde p e q sao primos Grandes

e eh um primo relativo de (p-1)(q-1)

calcular numeros primos

-crivo

-miler rabin

usa o teorema de fermat

a^p-1 = 1(mod p)

se p eh primo entao eh satisfeito para todo a

porem pode ocorrer falso positivo , dizer q um numero eh primo quando na vdd nao eh

parte de otimiza;ao estudar pelo corman ,ou nao

Indução Matemática – Fórmula

• Divisão e Conquista (divide and conquer)

• Algoritmos Gulosos (greedy)

• Algoritmos de Tentativa e Erro (backtracking)

• Programação dinâmica (dynamic programming)

• Algoritmos de aproximação (approximation)

Questoes de prova:

-Teoria dos numeros

-Criptografia

-Big Int

- Heuristica, aproximacao(Conceito)

- p/np

- Talvez saber o que cada algoritmo faz…