PAA

JEFERSON GONÇALVES NORONHA SORIANO - 471110

A) VERDADEIRO. $F(n) \in O(G(n))$ ABUSANDO DA LINGUAGEM PODEMOS SUBSTITUIR $\in$ POR $=$, $F(n) = O(G(n))$ SIGNIFICANDO QUE $F(n)$ PODE SER MENOR OU IGUAL A $G(n)$, USANDO A DEFINIÇÃO DE O, TEMOS

$F(n) \leq c.G(n)$

SUPONDO $c = 1$ E $n = 4$

$F(4) \leq 1.G(4)$

$F(4) \leq G(4)$

OU SEJA $F(n)$ SÓ PODE SE NO MAXIMO IGUAL A $G(n)$

A IMPLICAÇÃO É QUE $G(n) \in O(F(n))$, TEMOS

$G(n) \leq c.F(n)$

SUPONDO $c = 1$ $n = 4$

$G(4) \leq 1.F(4)$

$G(4) \leq F(4)$,

OU SEJA $G(n)$ SO PODE SER NO MAXIMO IGUAL A $F(n)$,

ASSIM PROVANDO QUE É VERDADEIRO

B)

FALSO, O Θ DIZ QUE AS FUNÇOES CRESCEM DE FORMA EQUIVALENTE ENTÃO $F(n) + G(n) = \Theta(\min(F(n), G(n)))$, SE UM LADO CRESCE DE FORMA QUADRÁTICA O OUTRO LADO TAMBEM TEM QUE CRESCER DE FORMA QUADRATICA, ABUSANDO DA LINGUAGEM, VAMOS SUBSTITUIR O $\in$ POR $=$, FICANDO EQUIVALENTE, ASSIM TENDO, $F(n) + G(n) = \Theta(\min(F(n), G(n)))$

USANDO A DEFINIÇÃO DE Θ

$$c \cdot \min(F(n), G(n)) \leq F(n) + G(n) \leq c \cdot \min(F(n), G(n))$$

SUPONDO $c = 1$, $n = 4$

$$1 \cdot \min(F(4), G(4)) \leq F(4 + F(4) \leq 1 \cdot \min(F(4), G(4))$$

$$1 \cdot 4 \leq 4 + 4 \leq 1 \cdot 4$$

$$4 \leq 8 \leq 4$$, TORNANDO FALSA, JA QUE NÃO SÃO IGUAIS.

c) VERDADEIRO, TEMOS $F(n) \in O(g(n)) \rightarrow 2^{F(n)} \in O(2^{g(n)})$, NA PRIMEIRA PARTE DA IMPLICAÇÃO DIZ QUE $F(n) \leq G(n)$, E NA SEGUNDA PARTE ELEVAMOS 2 A ESSA FUNÇÕES TENTO $2^{F(n)} \in O(2^{G(n)})$, USANDO A DEFINIÇÃO DE O, TEMOS:

$$2^{F(n)} \leq c \cdot 2^{G(n)}$$

SUPONDO $c = 1$ E $n = 4$

$$2^{F(4)} \leq 1 \cdot 2^{G(4)}$$

$2^{F(4)} \leq 2^{G(4)}$, ENTÃO $F(n)$ SO PODERAR SER MENOR OU IGUAL A $G(n)$, O MESMO VALE PARA $2^{F(n)} \leq 2^{G(n)}$

D) VERDADEIRO. $F(n) \in O(F(n)^2)$, DIZ QUE $F(n)$ É MENOR OU IGUAL A $F(n)^2$, USANDO A DEFINIÇÃO DO O, TEMOS

$$F(n) \leq c \cdot F(n)^2$$

SENDO $c = 1$, $n = 4$

$$F(4) \leq 1 \cdot F(4)^2$$

$$F(4) \leq F(4) \cdot F(4)$$

$$4 \leq 4 \cdot 4$$

$4 \leq 16$, $F(n)$ SEMPRE SERA MENOR OU IGUA $F(n)^2$

02)

```
MULTIMATRIZ(A[][], K):      //A nxn
    SE K == 1:
        RETORNA A

    AUX <- MULTIMATRIZ(A, K/2)

    C <- ALOCA MATRIZ(n)(n)

    PARA i DE 0 ATE n:
        PARA J DE 0 ATE n:
            PARA X DE 0 ATE n:
                C[i][J] += AUX[i][X] * AUX[X][J]

            SE K%2 == 1:      //SE FOR IMPAR
                PARA X DE 0 ATE n:
                    C[i][J] += AUX[i][X] * A[X][J]

    RETORNA C
```

ANALISANDO O PIOR CASO

ANALISANDO O PIOR CASO

VAMOS COMEÇAR ANALISANDO OS LAÇOS DENTRO DA FUNÇÃO, TEMOS UM LAÇO ANINHADO COM OUTRO LAÇO QUE ESTA ANINHADO COM OUTRO LAÇO MAIS OUTRO LAÇO SE O QUE ESTAMOS ANALISANDO É IMPAR E NO PIOR CASO SEMPRE ENTRA, TODOS OS LAÇOS VARIAM EM FUNÇÃO DE n, OS LAÇOS QUE ESTÃO ANINHADOS, MULTIPLICAMOS UM POR OUTRO ENTÃO TEMOS

$$n \cdot n \cdot (n+n)$$

$$n \cdot n \cdot 2n$$

$$\boxed{2n^3}$$, TEMOS $O(n^3)$

AGORA VAMOS ANALISAR A CHAMADA RECUSIVA SOMANDO COM $O(n^3)$, TEMOS

$$\begin{cases} T(1) = 1 \\ T(n) = T(n/2) + n^3 \end{cases}$$

T(n/2) É A CHAMADA RECUSIVA E O n^3 É A ANALISE DE COMPLECIDADE QUE ACABAMOS DE CALCULAR, DOS LAÇOS ANINHADOS, VAMOS CALCULAR

$$T(n) = T(n/2) + n^3$$

$$T(n) = T(T(n/4) + (n/2)^3) + n^3$$

$$T(n) = T(T(T(n/8) + (n/4)^3) + (n/2)^3) + n^3 = 2^k(n/2^k) + 3n^3$$

$$T(n) = T(T(T(T(n/16) + (n/8)^3) + (n/4)^3 + (n/2)^3)) + n^3 = 2^k(n/2^k) + 4n^3$$

$$\vdots$$

$$2^k F(n/2^k) + k \cdot n^3$$

$$n/2^k = 1$$

$$n = 2^k$$

$K = \log_2 n$, ENTÃO $2^{\log_2 n} F(1) + \log_2 n \cdot n^3$

$$n \cdot 1 + \log_2 n \cdot n^3$$

$$\boxed{n + \log_2 n \cdot n^3}$$

LOGO TEMOS QUE A COMPLEXIDADE É $O(n^3 \cdot \log n)$

03) FAIXMAXIMA(A, n):

```
VEC ← ALOCA VETOR (n)

PARA X DE 0 ATÉ n:
    VE[X] = Apx + Acx   //VEC VAI RECEBER A SOMA DE GOLS FEITOS
                                          +
                                        GOLS SOFRIDOS
                                        (A DUPLA (Px, Cx))

MAXIMO ← 0
INICIO ← 0
FIM ← -1

i ← 1

SOMA ← 0
PARA F ← 1 ATÉ n:
    SE (SOMA + VEC[F] < 0):
        i ← F + 1
        SOMA ← 0
    SENÃO:
        SOMA += VEC[F]

    SE (SOMA > MAXIMO):
        MAXIMO ← SOMA
        INICIO ← i
        FIM ← F

RETORNA INICIO, FIM
```

ANALISE DE PIOR CASO:

ANALISE DE PIOR CASO:

VAMOS ANALISAR OS CICLOS PORQUE ELE VAI TER A MAIOR INFLUENCIA NA ANALISE DE COMPLECIDADE, TEMOS 2 CASOS NÃO ANINHADOS QUE VARIAM EM TORNO DE n, ENTÃO

$n + n$

$2n$, LOGO A COMPLEXIDADE É $\underline{O(n)}$

04) MAIORSUBPOLINDROMA(X, n)

```
    VEC ← ALOCA MATRIZ(n+1)(n+1)

    AUX ← STRING

    PARA K DE 0 ATE n:
        AUX[K] ← X[n-K-1]    //AUX STRING REVERSA DA STRING X


    PARA i DE 0 ATÉ n:
        PARA J DE 0 ATÉ n:
            SE (i==0 OU J==0):
                VEC[i][J] ← 0

            SE NÃO SE(X[i-1] == AUX[J-1]):
                VEC[i][J] ← VEC[i-1][J-1] + 1

            SENÃO:
                VEC[i][J] ← MAX(VEC[i-1][J], VEC[i][J-1])


    RETORNA VEC[n][n]
```

ANALISE DE PIOR CASO

ANALISE DE PIOR CASO

VAMOS ANALISAR OS LAÇOS TEMOS UM LAÇO QUE VARIA EM FUNÇÃO DE n, DEPOIS TEMOS 2 LAÇOS ANINHADO VARIANDO EM FUNÇÃO DE n, QUANDO É ANINHADO MULTIPLICAMOS $n \times n$, ENTÃO TEMOS

$n + n \cdot n$

$n + n^2$, LOGO A COMPLEXIDADE DE $O(n^2)$, PORQUE PEGAMOS O MAIOR GRAU

05)

```
CONSTRUIRPALAVRA(S, n):
    VEC ← ALOCA VETOR(n)
    VEC[0] ← DICT(S[0])    //FUNÇÃO QUE RETORNA VERDADEIRO SE A PALAVRA
                           TIVER NO DICIONARIO E FALSO CASO NÃO

    PARA i DE 1 ATE n:
        q ← FALSO
        PARA J DE 1 ATE i:
            SE(VEC[J-1] E DICT(S[J])):
                q ← VERDADEIRO

            VEC[i] ← q

    RETORNA VEC[n]
```

ANALISE PIOR CASO

ANALISE PIOR CASO

VAMOS ANALISAR OS LAÇOS, PORQUE OS CICLOS É O QUE MAIS INFLUENCIA A COMPLEXIDADE, TEMOS 2 LAÇOS ANINHADOS EM FUNÇÃO DE n, LOGO

$$n \cdot n$$

n^2, ENTÃO TEMOS COMPLEXIDADE $O(n^2)$