PAA

NOME: JEFERSON GONÇALVES NORONHA SORIANO

MATRICULA: 471110

01) A) FREQUENCIA: (9, 6, 6)

B) NÃO PODE FORMAR PORQUE TANTO O 0 E 00 EM BIT REPRESENTA A MESMA COISA, LOGO TERIA O MESMO CODIGO PARA A MESMA LETRA

C) FREQUENCIA: (2, 3, 9)

D)

02) A) LUCRO():

V ← RECEBE OS VALORES DE DEALINE

L ← RECEBE OS VALORES DE LUCRO

// ASSOCIA O DEADLINE ESPECIFICO A Um LUCRO
EX: VALOR DEADLINE 3 TEM LUCRO 20

ORDENA() // VAI ORDENAR O DEADLINE E LEVAR OS LUCROS TAMBEM PARA NÃO PERDER A REFERENCIA DO DEADLINE

n ← ALOCA VETOR COM O TAMANHO DE |V|

A ← {1}

i ← 1

~~VEC ALOCA~~

PARA m DE 2 ATE n:

SE $V_i == V_m$: // DEADLINE IGUAIS

q ← MAX(L_i, L_m) // MAIOR LUCRO

A ← A ∪ {q} // FAZ UNIÃO ENTRE O NOVO LUCRO + O QUE TINHA ANTES

i ← m

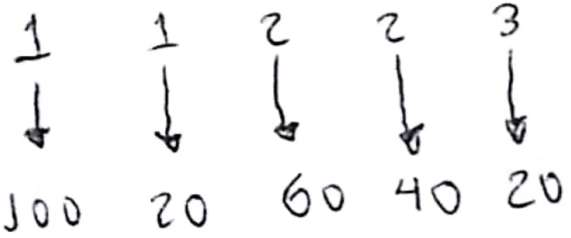
SENÃO: // SE OS VALORES NÃO SÃO IGUAIS ENTÃO COLOCA O VALOR DO LUCRO DE DEADLINE ESPECIFICO

A ← A ∪ {L_m}

i ← m

RETORNA A

B) COMO VOU ORDENAR O DEADLINE DOS MENORES PARA OS MAIORES SEM PERDER A REFERENCIA DO LUCRO, ASSIM POSSO ORDENAR EM ORDEM CRESCENTE, ENTÃO GARANTO SEMPRE O MAIOR LUCRO SE A DEADLINE FOR IGUA, O ALGORITMO ESCOLHE O MAIOR LUCRO COM AS DEADLINE IGUAIS, SE NÃO TIVE, ENTÃO JA VAMOS TAR COM O MAIOR LUCRO, EXPLICANDO O PORQUE ORDENAR A DEADLINE EM ORDEM CRESCENTE, E PORQUE GARANTE QUE NÃO VAMOS TER DEADLINE CONFLITANTES, EXEMPLO:



OS DEADLINE 1 PEGAMOS A DE MAIOR LUCRO, 100

OS DEADLINE 2 PEGAMOS A DE MAIOR LUCRO, 60

A DEADLINE 3, COMO É A ÚNICA É IGUAL A 20, LOGO TEMO 100 + 60 + 20

03) A)

```
DFS-VISITE(G, U):
    U.COR ⇦ CINZA
    PARA CADA V ∈ G.ADJ[U]:
        SE V.COR == CINZA
            RETORNA V

        SE V.COR == BRANCO
            DFS(G, V)

    U.COR = PRETO
```

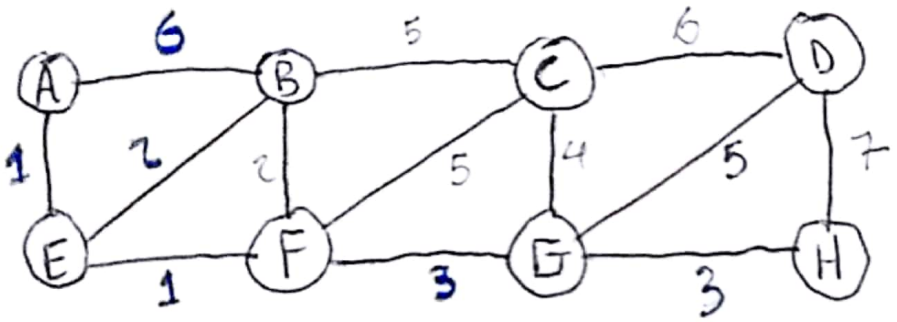
```
DFS(G, V):
    PARA CADA U ∈ G:
        U.COR ⇦ BRANCO
    PARA CADA U ∈ G:
        V = DFS-VISITE(G, U)

    SE V != NULO:
        RETORNA V

    SENÃO:
        RETORNA NULO
```

O CODIGO FAZ TODO MUNDO RECEBER BRANCO COM ESTADO INICIAL, QUE É IGUAL NÃO VISITADO, QUANDO PASSA POR VERTICE O ESTADO/COR MUDA PARA CINZA, QUE QUER DIZER QUE O VERTICE JA FOI VISITADO, QUANDO A COR É BRANCA CHAMAMOS O DFS QUE VER SE JA VISITAMOS AQUELE VERTICE MAS QUANDO O DFS - VISITE É CHAMADO PELO DFS ELE VERIFICA QUE É CINZA SE ISSO ACONTECER É PORQUE ELE JA FOI VISITADO ANTES, E ISSO SO ACONTECE QUANDO TEMOS UM CICLO, ENTÃO REMOVEMOS A ARESTA COMO É UM CLICLO O GRAFO CONTINUA CONEXO, SO PRECISAMOS FAZER ISSO UMA VEZ

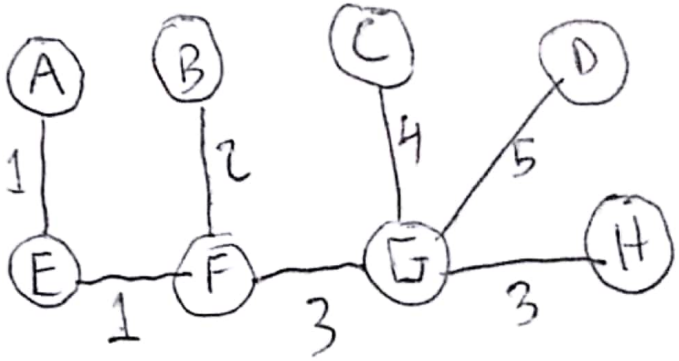
B) A B TEM POUCA ALTERAÇÃO DIANTE DO CODIGO ANTERIOR, UNICA COISA QUE VAI MUDAR É QUE NO PRIMEIRO CODIGO É QUE ELE ENCONTRAVA UM VERTICE E JA PARAVA, NESSE VAI CONTINUAR RETIRANDO OS CICLOS ATE NÃO PODER MAIS RETIRA E QUE ELE FIQUE CONEXO, ENTÃO BASICAMENTE VAMOS REMOVER ATE GERAR UMA ARVORE MINIMA E ARMAZENAR EM UM CONTADOR, QUANTOS VERTICES RETIRAMOS NO TOTAL, NO CODIGO MESMO VAMOS CRIAR A VARIAVEL CONT E NO IF DO DFS, NO SE V != NULO NO LUGAR DE RETORNAR V, VAMOS DELETAR G[V] E CONT++

04)

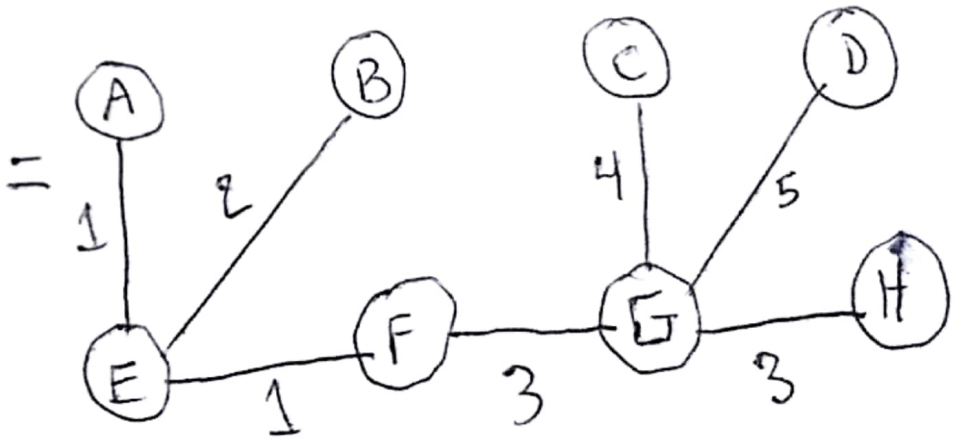
A) 19



=

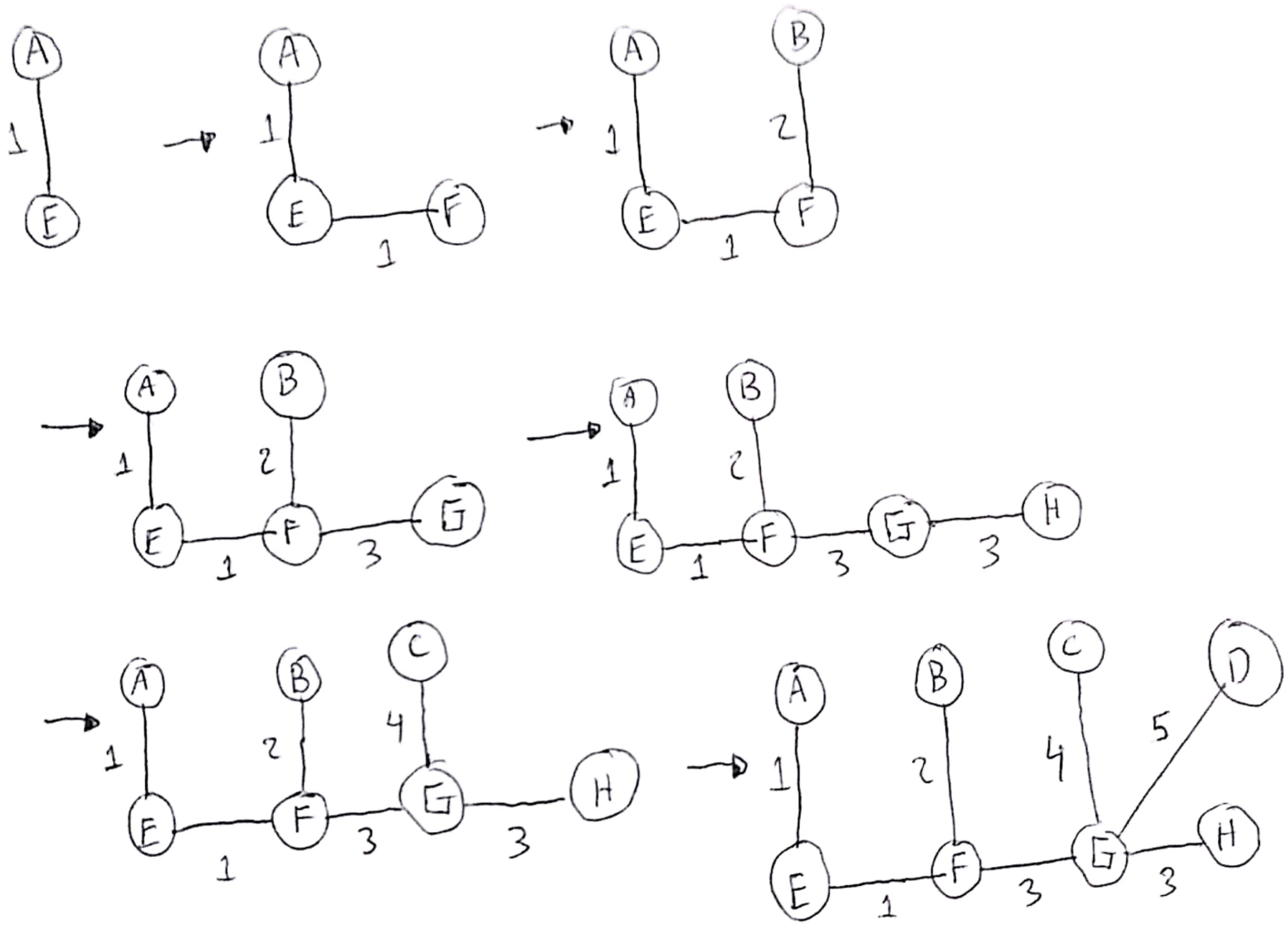


=



B) 2

c)



SEGUINDO O ALGORITMO KRUSKAL, VAMOS MONTAR A ARVORE GERADORA MINIMA SEGUINDO A LOGICA DE PEGAR ~~O VERTICE~~ A ARESTA DE MENOR PESO

UM CORTE QUE JUSTIFICA É O CORTE