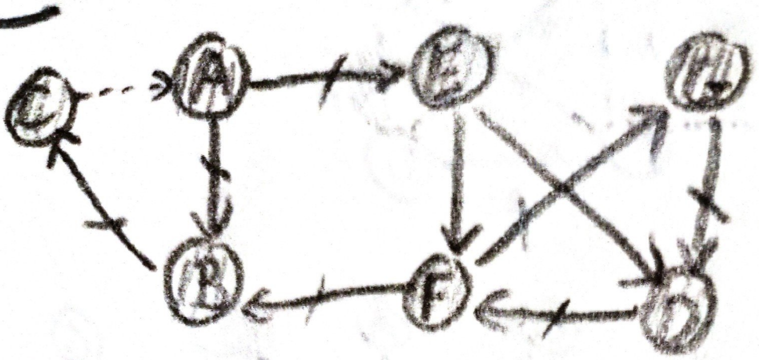
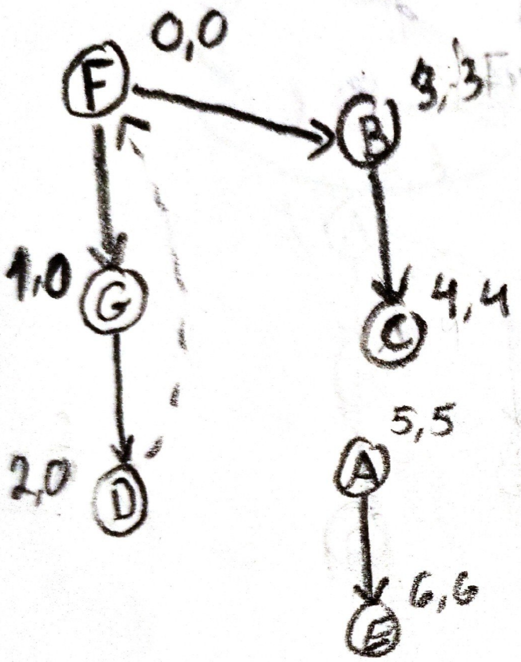
ER 2022

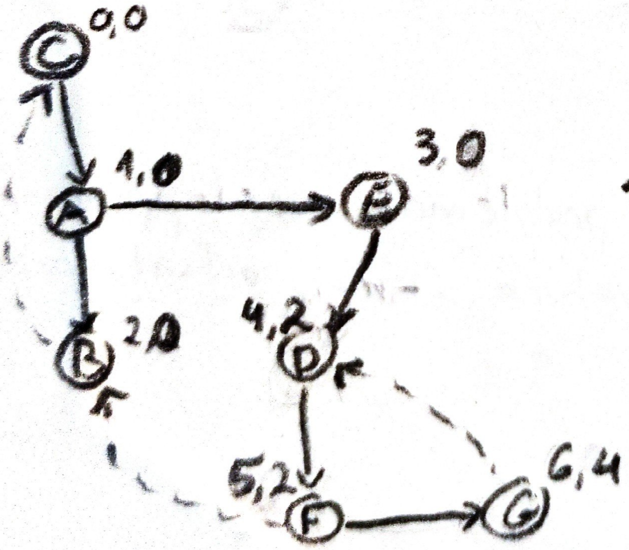


ⓐ



{C}
{B}
{D, G, F}
{E}
{A}

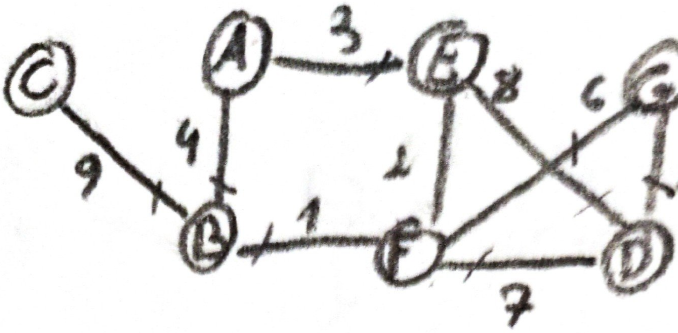
ⓑ Basta adicionar {C, A}, fazendo a DFS tree de novo, começando no C, obtém-se:

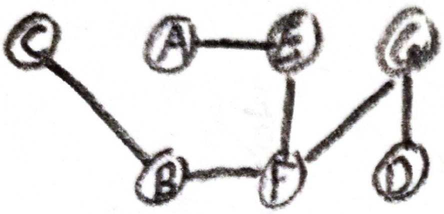


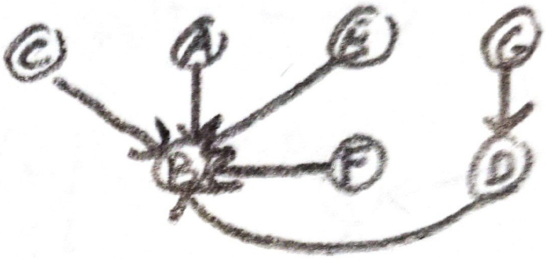
{C, A, E, B, D, F, G}

2

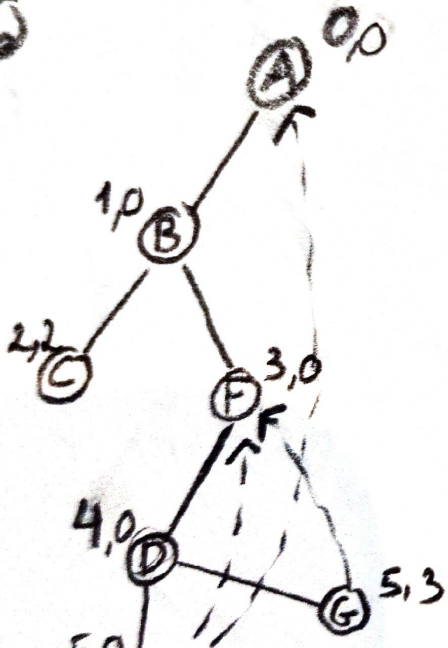
a







b



APs : {B}

c

Muitas recomputações, há problemas overlapping e há a propriedade da substrutura ótima, portanto podemos usar um approach de DP.

→

④ DP → Top-Down $A = (a_1, \dots, a_n)$

```
Function f(idx, sm1, sm2):
    if dp[idx, sm1, sm2] is cached:
        return dp[idx, sm1, sm2]
    if sm1 = 0 ∧ sm2 = 0:
        return True
    if idx = 0 ∧ (sm1 ≠ 0 ∨ sm2 ≠ 0)   } Redundante
        return False
    if idx = n:
        return sm1 = sm2 = sum(A)/3
    poss1 = f(idx - 1, sm1 - A[idx], sm2)
    poss2 = f(idx - 1, sm1, sm2 - A[idx])
    poss3 = f(idx - 1, sm1, sm2)
    return dp[idx, sm1, sm2] = poss1 or poss2 or poss3
```

→ Apesar da complexidade temporal e espacial ser $O(nS^2)$, uma vez que há $n \times \frac{S}{3} \times \frac{S}{3}$ estados, este problema é apenas resolvido em tempo pseudo-polinomial, uma vez que se considera que se considera que a relação entre n e S é exponencial.

→ Está correto pois preenche a tabela dp, para todos os subproblemas

$dp[i][sm1][sm2] = True \iff$ é possível fazer 2 subconjuntos cujas somas são sm1 e sm2 com os i primeiros elementos.

Portanto $f(len(A), \frac{sum(A)}{3}, \frac{sum(A)}{3})$ retorna a resposta ao problema.