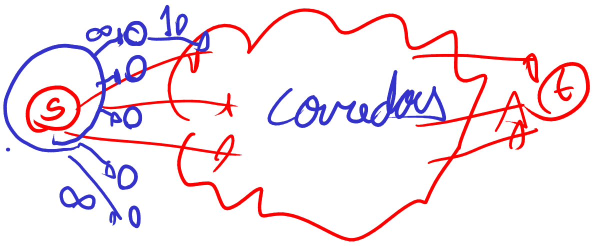
1. *(2.5pt)* Um grafo dirigido e ponderado $G = (V, A, w)$ que identifica as rotas de evacuação de um colégio, no qual V é o conjunto de salas, A é o conjunto de corredores e $w : A \rightarrow \mathbb{Z}^+$ é a função que mapeia a quantidade de pessoas que podem passar por cada corredor. Considere $t \in V$ como o vértice que identifica o ponto de saída. Considere também um conjunto de salas $E \subset V$ que é o conjunto de salas de origem das pessoas no colégio. Deseja-se desenvolver um algoritmo que receba o grafo G, quais vértices pertencem a E e qual o vértice t. O algoritmo deve determinar qual o número máximo de pessoas que saem de E e atingem a saída t. Com base nesse problema, crie um algoritmo para atendê-lo.



FM : Edmonds-Karp

Algoritmo Q3.1

Entrada: V, A, w, t, E

$V \leftarrow V \cup \{s\}$ — 1 u.t

foreach $u \in E$ do — $c.|E|$

  $A \leftarrow A \cup \{(s,u)\}$

  definir $w((s,u)) = \infty$

$G \leftarrow (V, A, w)$ — 1

$V_f \leftarrow V$ — 1

$A_f \leftarrow A$ — 1

$w_f \leftarrow w$ — 1

foreach $(u,v) \in A$ do — $c.|A|$

  $A_f \leftarrow A_f \cup \{(v,u)\}$

  definir $w_f((v,u)) = 0$

$G_f = (V_f, A_f, w_f)$ — 1

executar Algoritmo Edmonds-Karp(G, s, t, G_f) // Alg 23 + Alg 24 — $c.|V||A|^2$

$fm \leftarrow 0$ — 1

foreach $(u,v) \in A_f$ do — $2.c.|A|$

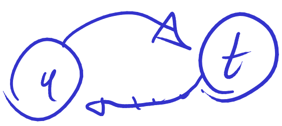
  if $v = t$ then

    $fm \leftarrow w_f((t,u)) + fm$

return fm — 1

$n = |V|$ $e = |E|$

$m\ |A|$

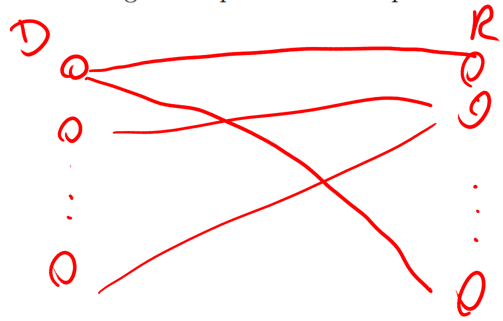


$e \leq n$

$T(n,m,e) = 8 + c.e + c.m + 2.c.m + c.n.m^2$

$T(n,m) \leq c_1 \cdot c.n.m^2$ $T(n,m,e) \in O(n.m^2)$

2. *(2.5pt)* Deseja-se desenvolver um algoritmo que identifique qual a quantidade máxima de doações que poderiam ser realizadas. O algoritmo recebe uma listagem D de doadores e uma listagem R de receptores. Também é recebido um conjunto composto por elementos (d, r) que indica que um doador d é compatível ao receptor r. Crie um algoritmo para atender o problema acima.



Algoritmo Q3.2

Entrada: $D, R, L = \{(d,r)\}$

1. $E \leftarrow \{\}$
2. Foreach $(d,r) \in L$ do
3. $\quad E \leftarrow E \cup \{\{d,r\}\}$
4. $V \leftarrow D \cup R$
5. $G \leftarrow (V, E)$
6. $M \leftarrow$ Hopcroft-Karp(G) // Alg. 27
7. return $|M|$

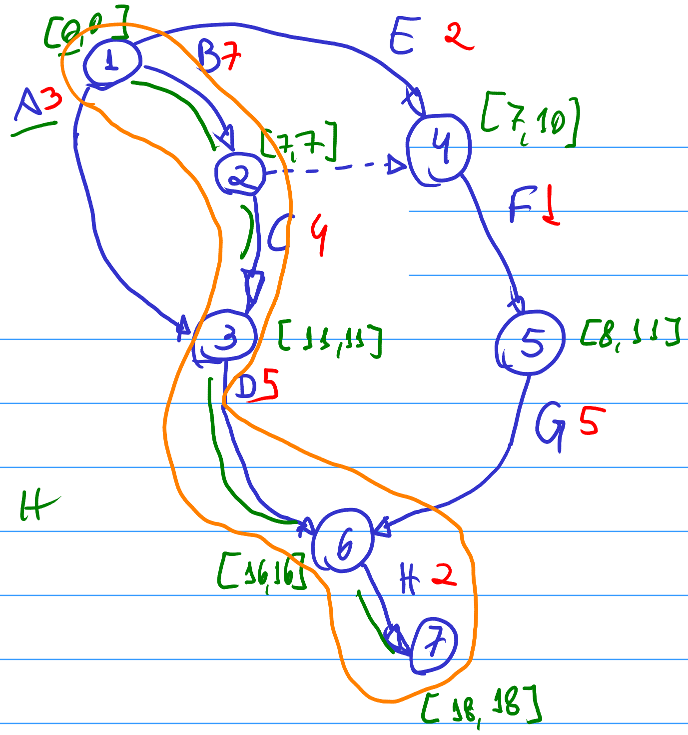
3. *(2.5pt)* Dado o conjunto de atividades abaixo e seus requistos, crie um grafo CPM e informe quais são as atividades críticas.

Atividade	Requisitos	Duração
A	-	3
B	-	7
C	B	4
D	A,C	5
E	-	2
F	B, E	1
G	F	5
H	D,G	2

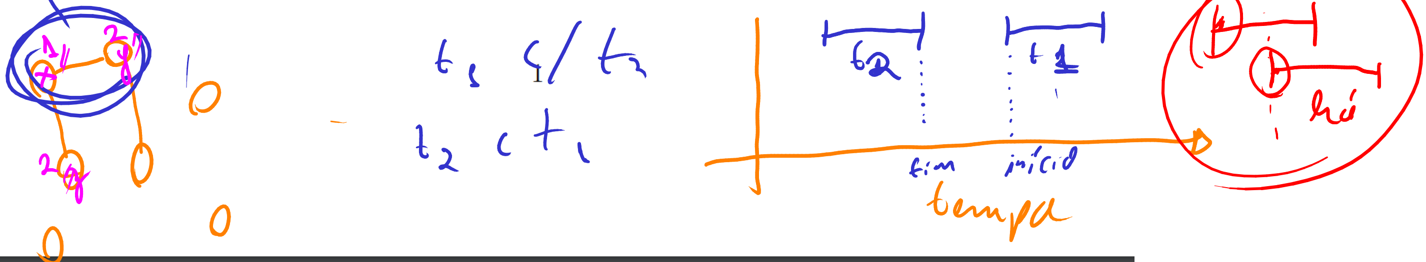
	E	T
1	0	0
2	7	7
3	11	11
4	7	10
5	8	11
6	16	16
7	18	18

Críticas:

B, C, D, H



4. *(2.5pt)* Dado um conjunto de turmas T, sendo que cada turma inicia aulas no horário $S_i \in \mathbb{Z}^+$ e termina no horário $F_i \in \mathbb{Z}^+$. Deseja-se saber qual a quantidade mínima de salas que podem ser



Not (Não há conflito sse (t_1 começa depois do fim t_2) OU t_2 começa depois do fim t_1))

alocadas, respeitando que cada sala só pode ser usada por uma turma em um determinado tempo. Especifique um algoritmo para resolver o problema.

Algoritmo Q3.4

Entrada: T, S_i, F_i

```
V ← T                                          → 1
E ← {}                                         → 1
foreach u ∈ V do                               — c.|V|
    foreach v ∈ V - {u} do                     → c.|V|
        if not( S_u > F_v ou S_v > F_u ) then  ] 1.u.t
            E ← E ∪ {{u,v}}
G ← (V, E)                                     — 1.u
salas ← Alg. Lawler (G)  //Algoritmo 34        → c.|V|.|E|.2,4423^|V|
return salas                                   → 1.u.t
```

(ñ há conflito)

$n = |T|$

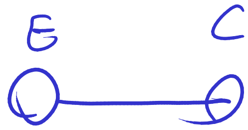
$m \leq |T|^2$

$$T(n,m) = 4 + c.n(c.n) + c.n.m.2{,}4423^n$$

$$T(n,m) \leq c_1.n.m.2{,}4423^n$$

$$\boxed{T(n,m) \in O(n.m.2{,}4423^n)}$$

3. *(2.5pt)* Em um sistema de entregas, há um conjunto de clientes C e um conjunto de entregadores E. Cada entregador $e \in E$ está disposto a realizar entregas nas localidades presentes no conjunto B_e. Cada cliente $c \in C$ mora na localidade identificada por L_c. Dados os conjuntos C, E, B_e para todo entregador $e \in E$ e a localidade L_c de cada cliente $c \in C$, deseja-se identificar qual entregador realizará cada entrega, maximizando o número de entregas.



Algoritmo Q3'3

Entrada: C, E, B_e, L_c

1. $V \leftarrow E \cup C$ → 1
2. $E \leftarrow \{\}$ → 1
3. Foreach $e \in E$ do — $c.|E|$
4. Foreach $c \in C$ do — $c.|C|$
5. if $L_c \in B_e$ then → árvore balanceada $\log_2 |C|$
6. $E \leftarrow E \cup \{\{e,c\}\}$ → 1.ut
7. $G \leftarrow (V,E)$ → 4u.t
8. $M \leftarrow$ Hopcroft-Karp(G) → $c.\sqrt{|V|}.|E|$
9. return M — 1

$c = |C|$

$e = |E|$

$$T(c,e) = 4 + c_1.e(c_1.c(\log_2 c + 1)) + c.\sqrt{c+e}\,.\,c.e$$

$$T(c,e) \in O\left(\sqrt{c+e}\,.\,c.e + c.e\log_2 c\right)$$