Estudos p/ 2ª prova

Questão 1 (20

Seja $G = (V, E)$ um grafo direcionado e não ponderado.

a) (10 %) Discorra sobre conectividade em grafos direcionados. Apresente todos os tipos de grafos além de indicar sobre identificá-los.

b) (10 %) Caso G não seja desconexo, projete um algoritmo para identificar os componentes fortemente conexos de G

a) Em grafos direcionados, a conectividade diz respeito à existência de caminhos entre pares de vértices, respeitando a direção das arestas. Existem três principais classificações em relação a conectividade, essas são:

- Grafo fortemente conexo:

Para quaisquer pares de vértices (u, v), existe um caminho de u para v e de v para u, ou seja, $\hat{\Gamma}^{+}(u) = V$ e $\hat{\Gamma}^{-}(u) = V$

- Grafo semi-fortemente conexo:

Para quaisquer pares de vértices (u, v), há pelo menos um caminho entre eles (qualquer direção)

- Grafo fracamente conexo:

O grafo associado , ou seja, quando tiramos as direções do nosso grafo, é conexo. Porém, no grafo original não é possível alcançar todos os vértices, respeitando as direções.

Questão 2 (20 %)

Discorra sobre subgrafo e subgrafo induzido. Apresente exemplos de ambos os conceitos.

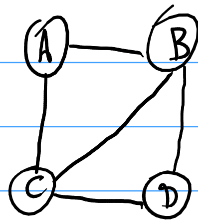
Seja $G = (V, E)$ um grafo e $H = (V', E')$ seu subgrafo, o subgrafo será válido somente se, $H \subseteq G$, $V' \subseteq V$ e $E' \subseteq E$

ex:

$G=(V,E)$ um grafo não direcionado

$V=\{A,B,C,D\}$

$E=\{\{A,B\},\{A,C\},\{B,D\},\{C,D\},\{B,C\}\}$

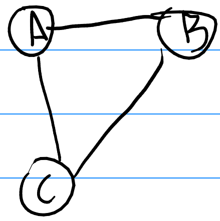


um subgrafo válido seria:

$H=(V',E')$

$V'=\{A,B,C\}$

$E'=\{\{A,B\},\{A,C\},\{B,C\}\}$

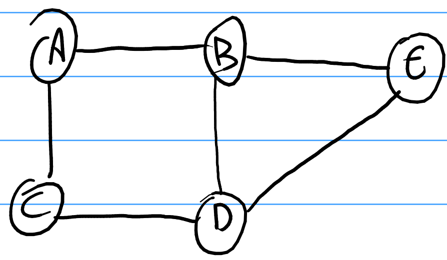


Um subgrafo é dado como induzido quando ao escolher o subconjunto de vértices V', o subconjunto de arestas E' será todas as arestas relacionadas em V'.

ex: $G=(V,E)$ um grafo não direcionado

$V=\{A,B,C,D,E\}$

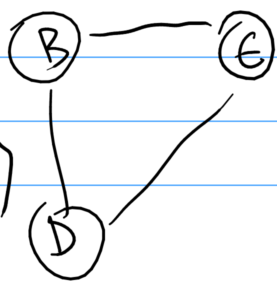
$E=\{\{A,B\},\{A,C\},\{C,D\},$
$\{B,D\},\{B,E\},\{D,E\}\}$



um subgrafo induzido válido seria:

$V'=\{B,D,E\}$

$E'=\{\{B,E\},\{B,D\},\{D,E\}\}$



Questão 3 (30 %)

Um *caminho crítico* é determinado pela identificação do maior período de atividades dependentes, além da consideração do tempo mínimo necessário para a realização de todas as tarefas (ou atividades). Uma aplicação interessante deste problema é a identificação do número mínimo de períodos para que um aluno possa se formar considerando todas as disciplinas de seu curso. Cumpre ressaltar que disciplinas que não estão na linha de pré-requisitos podem ser feitas em paralelo. **Modele** o problema do caminho crítico em grafos e **projete** um algoritmo para identificá-lo. Além disto, ilustre *um exemplo da sua modelagem.*

Podemos modelar o problema em grafos da seguinte forma:

Seja $G=(V,E)$ um grafo direcionado e acíclico (DAG) e as disciplinas serão os vértices e os pré-requisitos (dependências) serão as arestas. Para resolvermos o problema do caminho crítico utilizaremos uma ordenação topológica, isto é, para todo par de vértice (u,v), u tem que vir antes de v.
Segue o passo-a-passo do algoritmo:

1. Pegamos a base do grafo, ou seja, o vértice que possui grau de entrada $\emptyset$.

2. Adicionamos na lista de ordenação, e removemos um grau de entrada para todos os vizinhos.

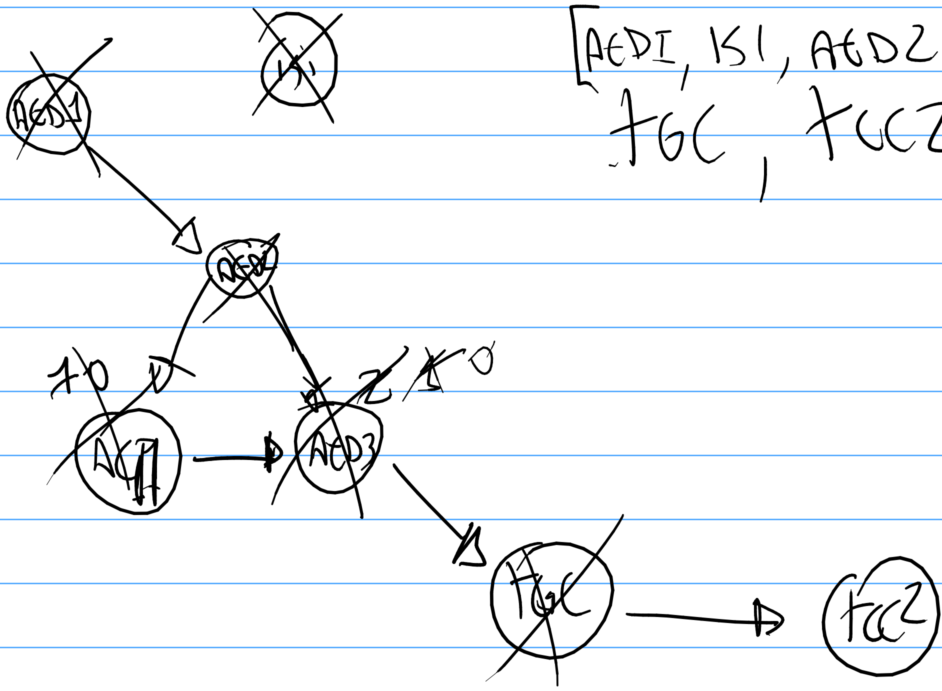
3. O vértice que chegou ao grau de entrada $\emptyset$, será adicionado na lista de ordenação

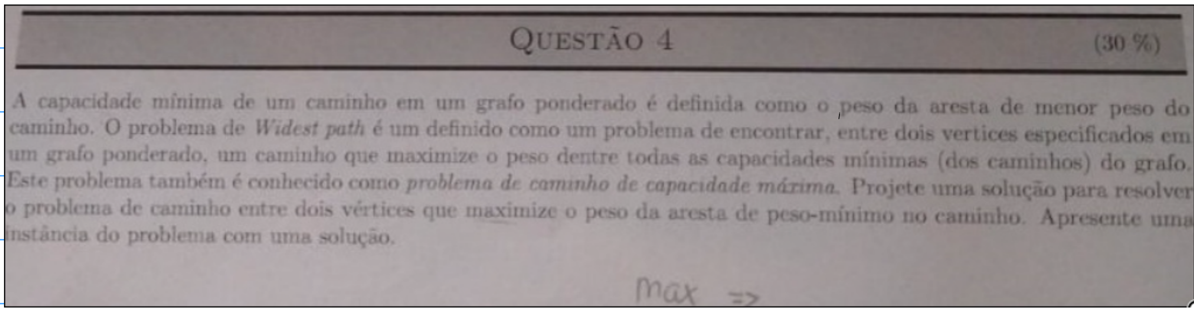
4. Repetir até que todos os vértices sejam processados

Agora que temos nosso grafo organizados em ordem topológica, podemos percorrer o grafo de forma que saberemos os tempos mínimos para se chegar em cada matéria.
Logo o caminho crítico será o que tiver maior tempo acumulado, ou seja, a maior sequência de disciplinas até a conclusão do curso.

ordenação topo
[AED1, ISI, AED2, ACI, AED III, TGC, TCC2]

ex:



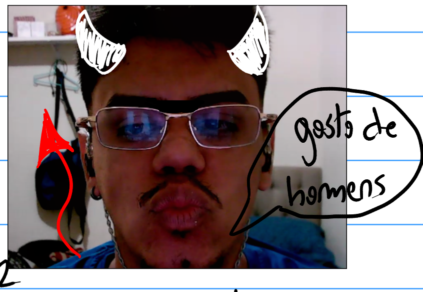


QUESTÃO 4 (30 %)

A capacidade mínima de um caminho em um grafo ponderado é definida como o peso da aresta de menor peso do caminho. O problema de *Widest path* é um definido como um problema de encontrar, entre dois vertices especificados em um grafo ponderado, um caminho que maximize o peso dentre todas as capacidades mínimas (dos caminhos) do grafo. Este problema também é conhecido como *problema de caminho de capacidade máxima*. Projete uma solução para resolver o problema de caminho entre dois vértices que maximize o peso da aresta de peso-mínimo no caminho. Apresente uma instância do problema com uma solução.

max =>

Questão 4:



Para resolvermos isso podemos usar a lógica do algoritmo de Dijkstra, só que de forma "inversa", segue o passo-a-passo:

1. Criar uma lista de capacidade:
   - Guardará o menor peso para cada vértice
   - capacidade[origem] = ∞ (não passou p/ aresta)
   - todos os outros vértices serão ∅

2. Criar uma fila de prioridade:

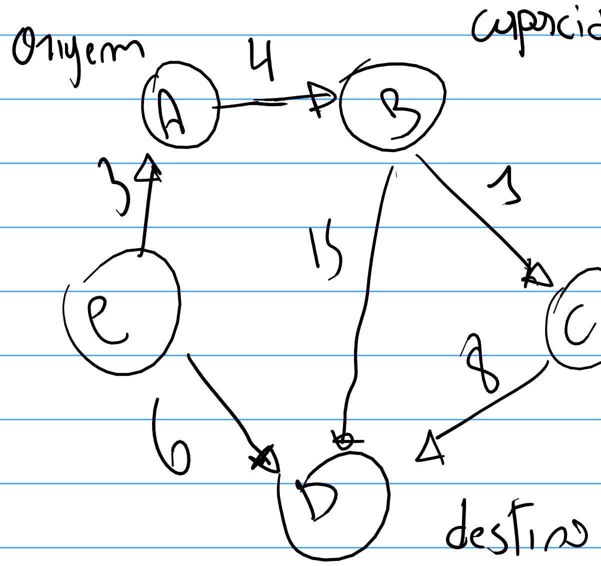
• Invés de ser a menor distância, será a maior capacidade
• A fila terá a maior capacidade conhecida

3. Enquanto a fila não estiver vazia:

I. Pegue vértice u com a maior capacidade
II. Para cada vizinho v de u:
• ver aresta com peso w entre u e v
• calcular o min(capacidade[u], w)
• Se a nova capacidade > capacidade[v], a capacidade[v] = nova capacidade
• Adicionar v na fila

4. Processar para todos os vértices e capacidade[destino] será o maior valor possível para o menor peso entre o caminho da origem até o destino.

ex:



capacidade = [A:∞, B:~~0~~ 4, C:~~0~~ 1, D:~~0~~ 4, E:0]

$\min(\infty,4)=4$

$4>0=4$

$\min(4,15)=4$

fila = [~~(A:∞)~~, ~~(B:4)~~),

Questão 1 (20 %)

Seja $G = (V, E)$ um grafo direcionado e não ponderado.

8 a) (10 %) Discorra sobre conectividade em grafos direcionados. Apresente todos os tipos de grafos além de indica[r] sobre identificá-los.

0 b) (10 %) Caso G não seja desconexo, projete um algoritmo para identificar os componentes fortemente conexos de [G]

a) Seja $G=(V,E)$ um grafo, a conectividade se dá quando há um caminho de um par de vértice (u,v). Há três principais tipos de conectividade, segue abaixo:

• Grafo fortemente conectado:

Para quaisquer pares de vertices (u,v), há um caminho de u para v e de v para u, isto é, $\Gamma^+(u) = V$ e $\Gamma^-(u) = V$.

• Grafo semi-fortemente conectado:

Para quaisquer pares de vértices (u,v), há pelo menos um caminho de u para v, ou de v para u.

• Grafo fracamente conexo:

Quando vemos o grafo associado, isto é, retiramos a direção do grafo, ele é conexo. Porém não conseguimos alcançar todos os vértices, respeitando as direções

Questão 2 (30 %)

Seja um grafo G com o seguinte conjunto de vértices $\{A, B, C, D, E, F\}$ e com a seguinte ponderação nas arestas.

	A	B	C	D	E	F
A	0	6	9	11	15	8
B	6	0	13	3	1	2
C	9	13	0	10	4	4
D	11	3	10	0	5	6
E	15	1	4	5	0	2
F	9	2	4	6	2	0

a) (10 %) Discorra sobre o algoritmo de Kruskal, além disto, encontre a árvore geradora mínima de G usando [o] algoritmo de Kruskal (mostre o passo-a-passo de sua solução)

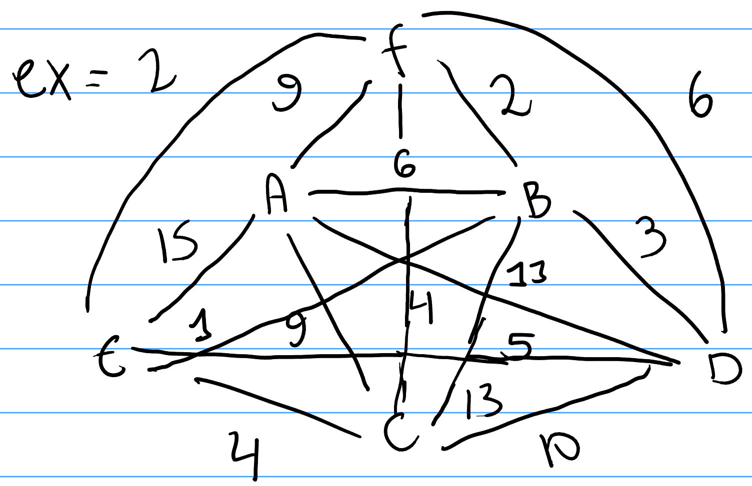
b) (10 %) Discorra sobre o algoritmo de Prim, além disto, encontre a árvore geradora mínima de G usando o algoritm[o] de Prim iniciando em F (mostre o passo-a-passo de sua solução)

c) (10 %) Discorra sobre o algoritmo de Dijkstra, além disto, encontre o menor caminho de F para todos os vértice[s] em G usando o algoritmo de Dijkstra.

a) Método para achar árvore geradora mínima (MST) de um grafo ponderado e conexo. Utiliza union-find para garantir que não haja ciclos:

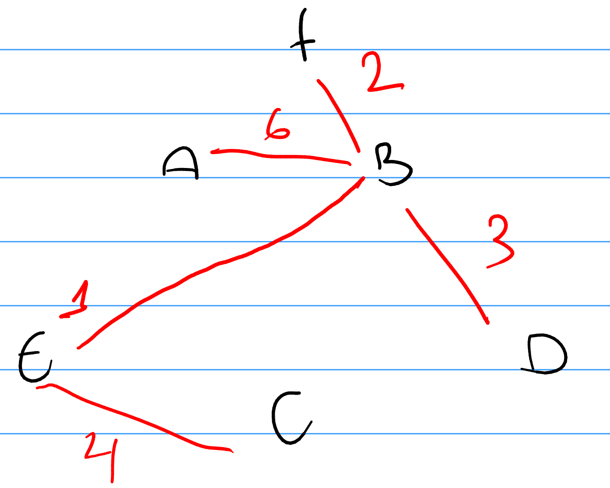
1. Ordena as arestas em ordem crescente de peso
2. Verifica se ao adicionar a aresta irá haver algum ciclo:
   - Se sim, passa para a próxima
   - Se não, adiciona na MST
3. O algoritmo para quando a árvore contém n-1 arestas

ex = 2



pesos_ordenados

[1,2,2,3,4,4,5,6,7,9,9,10,11,
13,15]



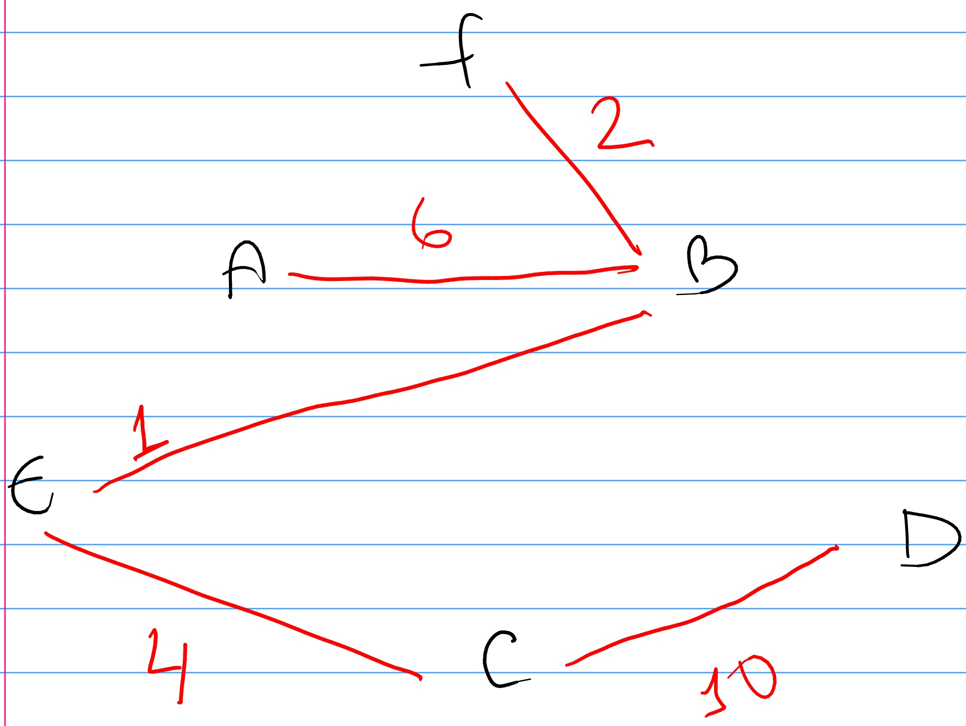
b) O prim funciona da seguinte forma:

1. Seleciona um vértice inicial para começar

2. Escolhe a aresta de menor peso, dos vizinhos da árvore atual. Se não formar ciclo adiciona na MST

3. termina quando a árvore tiver n-1 arestas

ex:



c) O algoritmo de Dijkstra serve para encontrar o menor caminho (pelo custo total dos pesos das arestas) até todos os outros vértices de um grafo ponderado e sem pesos negativos.

Passo-a-passo:

1. Inicialização
   - Define-se a distância do vértice de origem como ∅
   - todas as outras distâncias para os outros vértices são ∞

• Cria-se uma estrutura para saber quais vértices já foram visitados

2. Loço Principal

• Escolhe-se o vértice com a menor distância
• Marcamos esse vertice como "visitado"
• Para cada vizinho do vértice atual:
    • Calcula-se a distância total de origem até esse vizinho passando pelo vertice atual
    • Se essa distância for menor que registrada até agora, atualiza-se a distância

3. Repetimos até processar todos os vértices.

ex= distancias = [A: ~~∞~~ 9, B: ~~∞~~ 2, C: ~~∞~~ 6, D: ~~∞~~ 6, E: ~~∞~~ 2, F: ∅]

visitados = [F, A, B, C, D, E]

QUESTÃO 3 (25 %)

Explique, em detalhes, como calcular o fluxo máximo de uma rede. Apresente um exemplo que ilustre cada uma das etapas do algoritmo descrito.

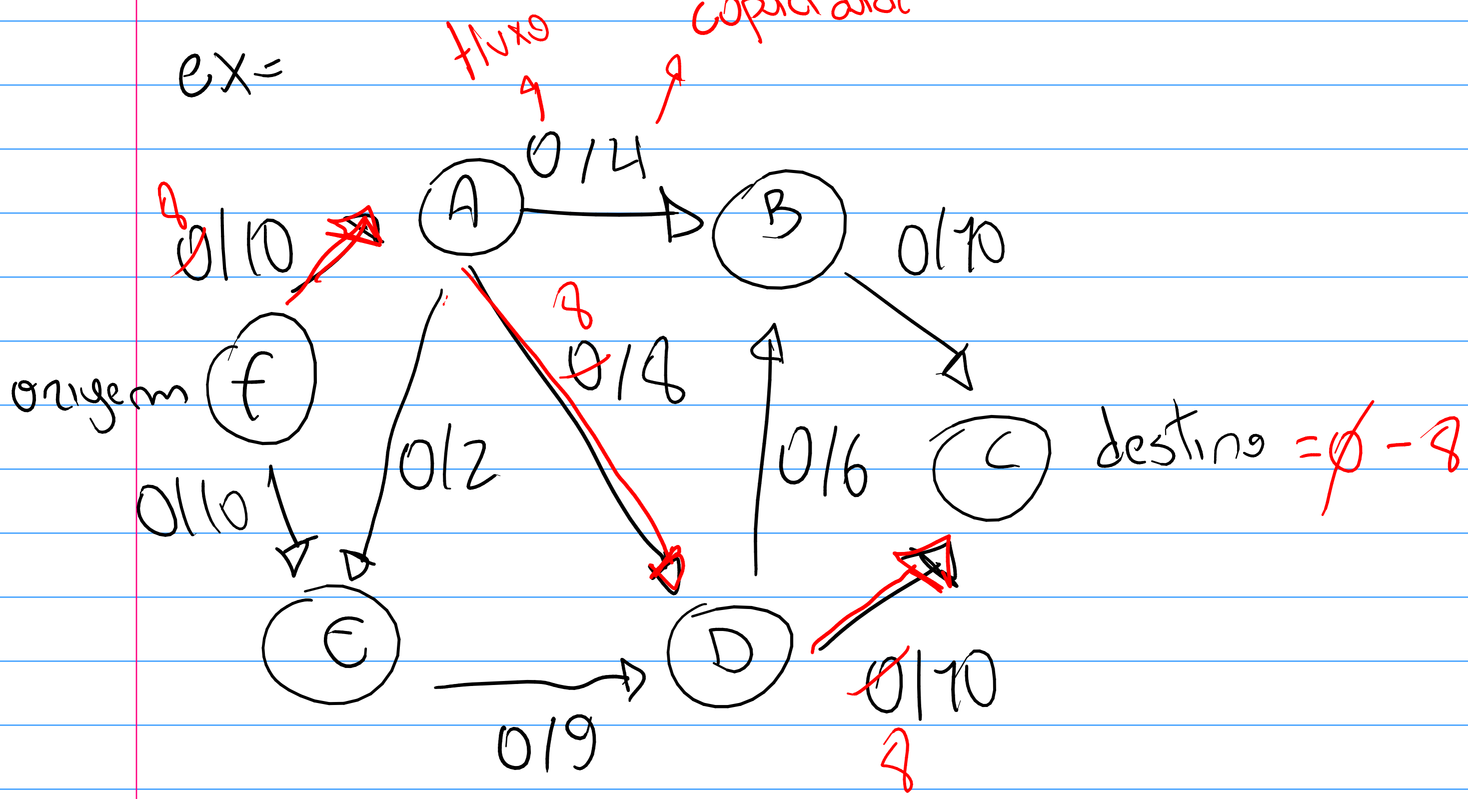
1. Começamos com o fluxo zero
2. Enquanto houver um caminho aumentante do nó origem até o destino com capacidade disponível:

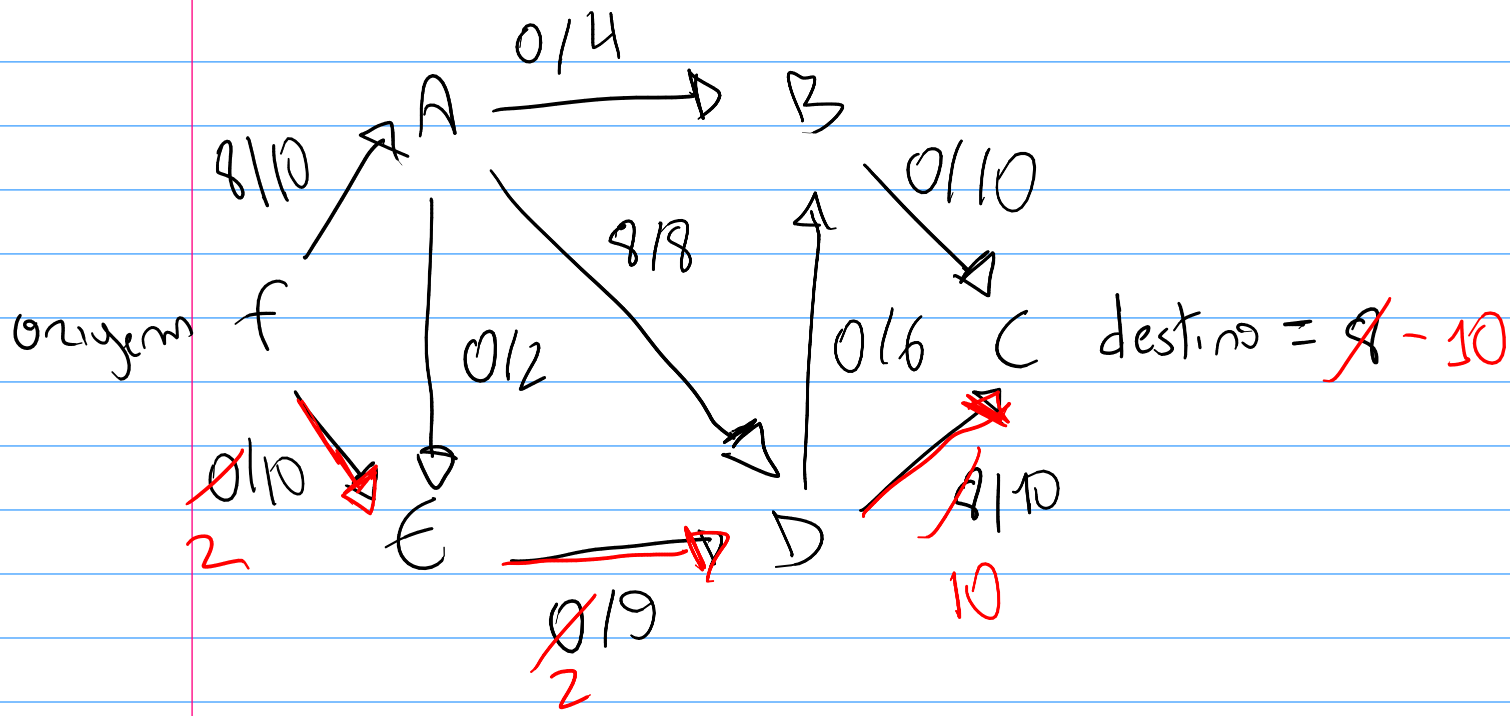
- Envia o máximo possível de fluxo por esse caminho (respeitando a capacidade mínima nas arestas do caminho)

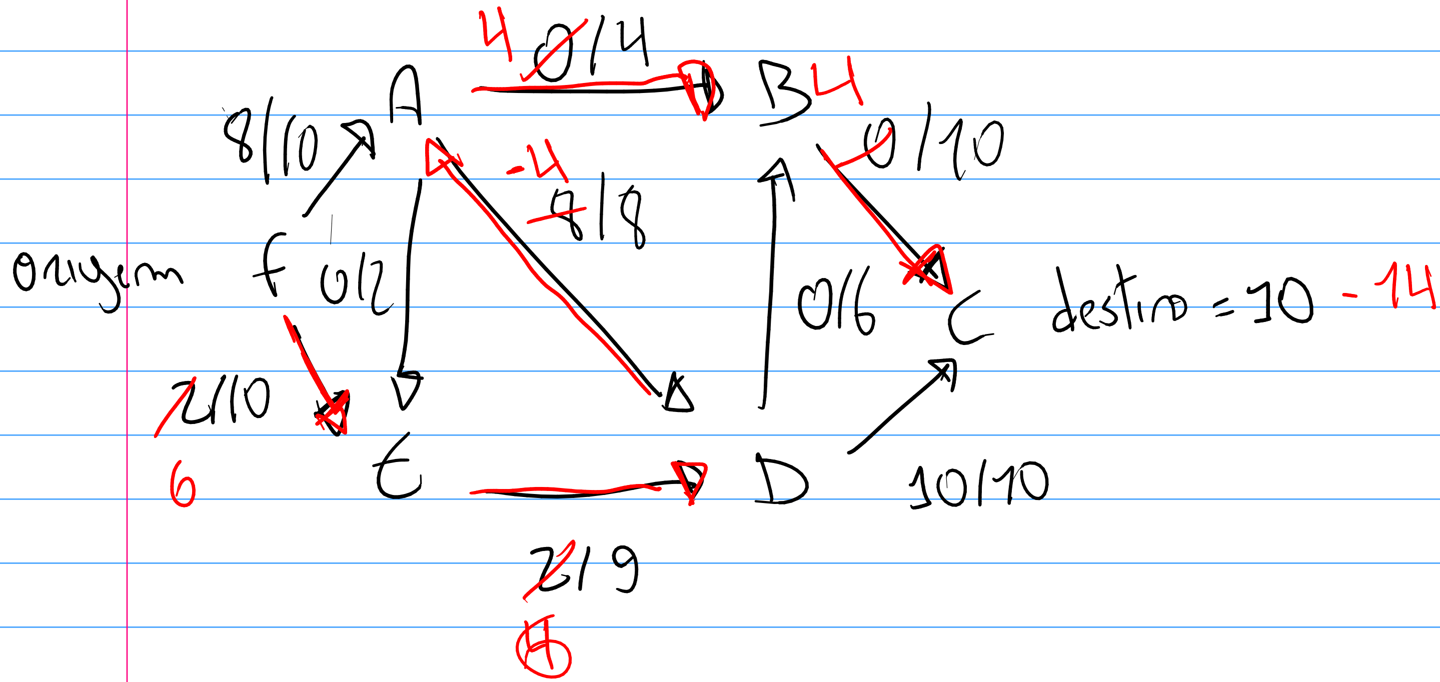
- Atualiza as capacidades:
  - Diminui a capacidade das arestas usadas
  - Cria ou aumenta a capacidade das arestas reversas (isso permite devolver fluxo se encontrar um caminho melhor depois)

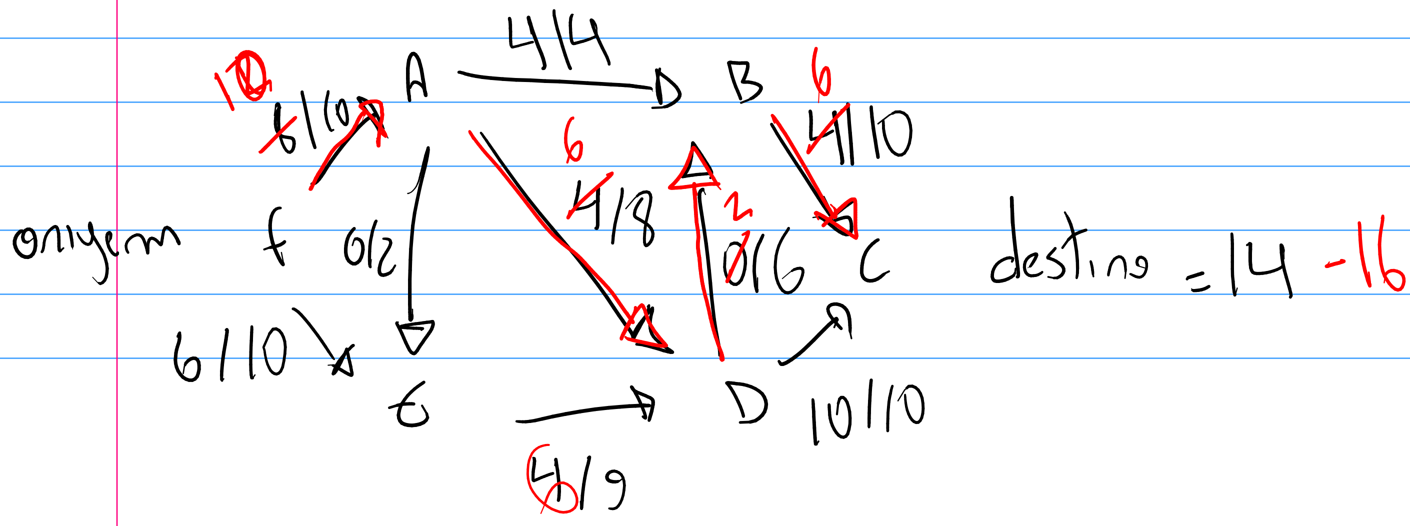
- Repete até algum caminho estar vazio p/ voltar ou cheio p/ seguir em frente.

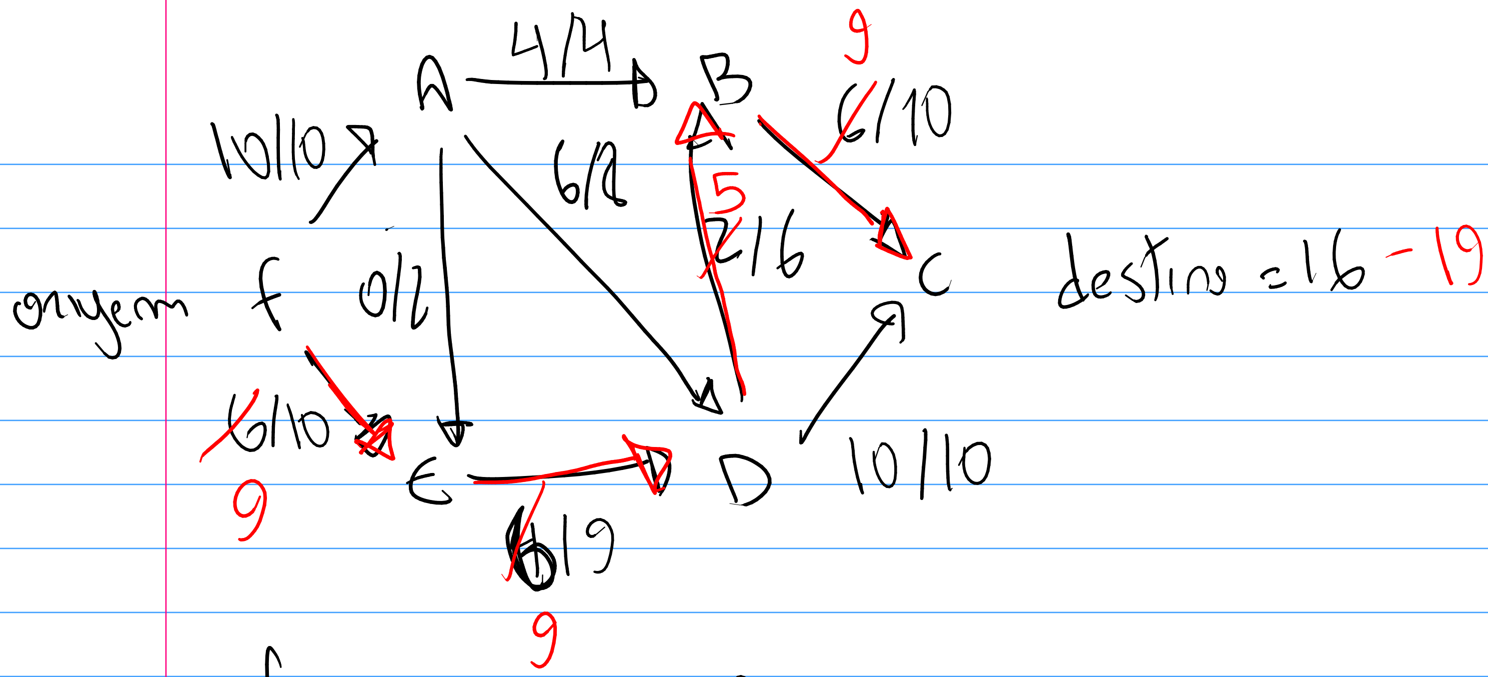
ex=




0/4
A
B
8/10
0/10
8/8
origem
f
0/2
0/6
C
destino = 8 - 10
0/10
2
E
D
8/10
10
0/9
2


4
0/4
A
B 4
8/10
-4
8/8
0/10
origem
f
0/2
0/6
C
destino = 10 - 14
2/10
6
E
D
10/10
2/9
4


4/4
10
A
B
6
6/10
4/10
6
4/8
2
origem
f
0/2
0/6
C
destino = 14 - 16
6/10
E
D
10/10
6
4/9



fluxo maximo = 19 //

Questão 4

(30 %)

A capacidade mínima de um caminho em um grafo ponderado é definida como o peso da aresta de menor peso do caminh
Sejam dois vértices quaisquer do grafo, projete uma solução para resolver o problema de caminho de capacidade máxin
entre estes vértices. Apresente uma instância do problema com uma solução.

Para resolvermos o problema de capacidade maxima, faremos o seguinte algoritmo:

1. Definir uma estrutura que defina as capacidades, sendo o vertice de origem marcado com ∞ e os demais com ∅

2. Criar uma fila de prioridade, que guardará a maior capacidade conhecida

3. Enquanto a fila não estiver vazia:

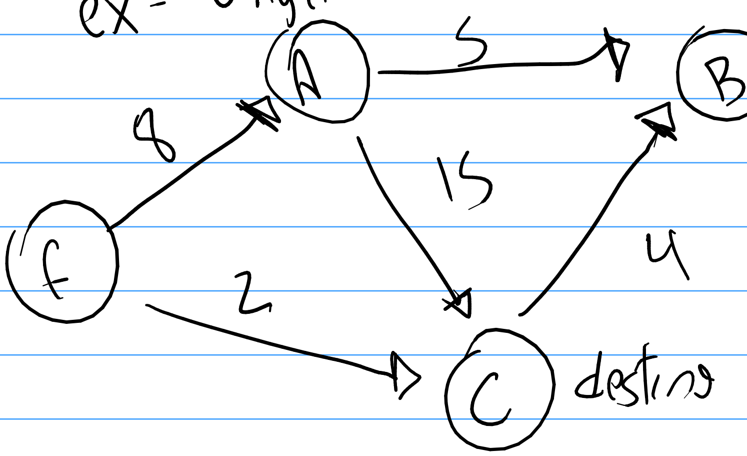
I. Pegar vértice u com a maior capacidade

II. Para cada vizinho v de u:

- Ver o peso w entre u e v
- Calcular uma nova capacidade = min(capacidade[u], w)
- Se nova capacidade > capacidade[v]:
  - capacidade[v] = nova capacidade
- Adicionar v na fila de prioridade

4. Processar todos os vértices e capacidade[destino] será a capacidade máxima.

ex = Origem



$\min(\infty, 15) = 15$

$15 > 0 = 15$

Capacidade = [A: ∞, B: ~~0~~ 5, C: ~~0~~ 15, F: 0]

fila = ((A: ∞),

Questão 1 (20 %)

Seja $G = (V, E)$ um grafo e N seu nome. Considere que os vértices do grafo sejam as letras do seu nome (caso haja letras iguais, mantenha a primeira ocorrência da letra e substitua as restantes pela letra seguida de um número). Por exemplo, $V = \{\{s\}, \{i\}, \{l\}, \{v\}, \{i1\}, \{o\}\}$

a) (10 %) Defina grafo completo, e mostre como ficaria o conjunto de arestas de G caso ele seja completo. Qual seria o número de arestas de um grafo completo de n vértices (apresente a fórmula)?

b) (10 %) Se G um grafo não-direcionado e completo. Defina grafo complementar, e mostre como ficaria o conjunto de arestas do grafo complementar de G. Qual seria o número de arestas de um grafo complementar G contendo n vértices?

a) $V = \{\{F\}, \{E\}, \{L\}, \{I\}, \{P\}, \{E1\}\}$

Seja $G=(V,E)$ um grafo direcionado, é dito completo K_n quando possui todas as arestas possíveis com n vértices. Dado por $\frac{n!}{2!(n-2)!} = \frac{n(n-1)\cancel{(n-2)!}}{2!\cancel{(n-2)!}} = \frac{n(n-1)}{2}$

Para o grafo $G=(V,E)$ e $V=\{f,e,l,i,p,e,1\}$ teremos $= \frac{6(5)}{2}$ arestas, logo 15 arestas.

b) O grafo complementar, denotado por $\bar{G}$, tem o mesmo conjunto de vértices V, e as arestas são as que faltam para tornar G em um grafo completo K_n. Logo, o grafo complementar do grafo G dado será um grafo nulo (sem arestas) pois G contém todas as possíveis arestas.

Para encontrar a quantidade de arestas do grafo complementar $\bar{G}$, temos:

$$\frac{n(n-1)}{2} - |E|$$

QUESTÃO 2 (30 %)

Seja $G = (V, E)$ um grafo direcionado e não ponderado.

a) (15 %) Discorre sobre conectividade em grafos direcionados. Apresente todos os tipos de grafos além de indicar sobre identificá-los.

b) (15 %) Caso G não seja desconexo, projete um algoritmo para identificar os componentes fortemente conexos de G

a) A conectividade de um grafo se dá quando temos caminhos entre pares de vértices, respeitando a direção. Temos três principais conectividade em um grafo:

- Grafo fortemente conexo

Para quaisquer par de vértices (u,v), há um caminho de u para v e de v para u. Ou seja, $\Gamma^+(u)=V$ e $\Gamma^-(u)=V$

- Grafo semi-fortemente conexo

Para quaisquer par de vertices (u, v), há pelo menos um caminho de u para v ou de v para u.

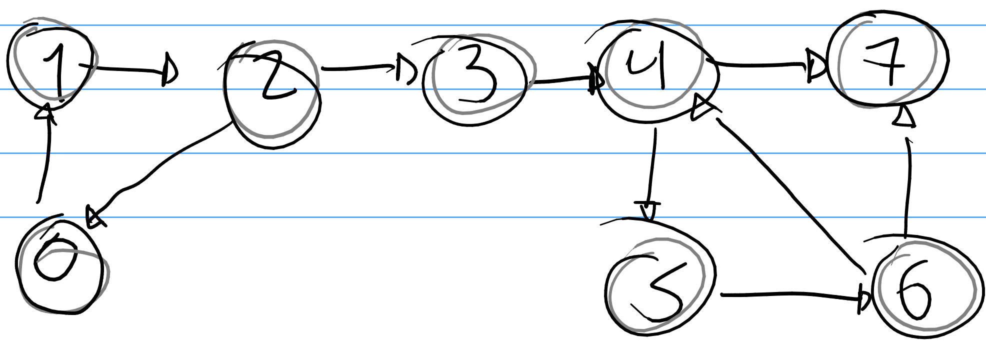
- Grafo fracamente conexo

Quando o grafo associado, ou seja, quando se ignora as direções do grafo original, é conexo. Porém, com o grafo original não é possivel alcançar todos os vertices, respeitando as direções.

b) Para encontrar os componentes fortemente conexos, faremos o seguinte algoritmo:

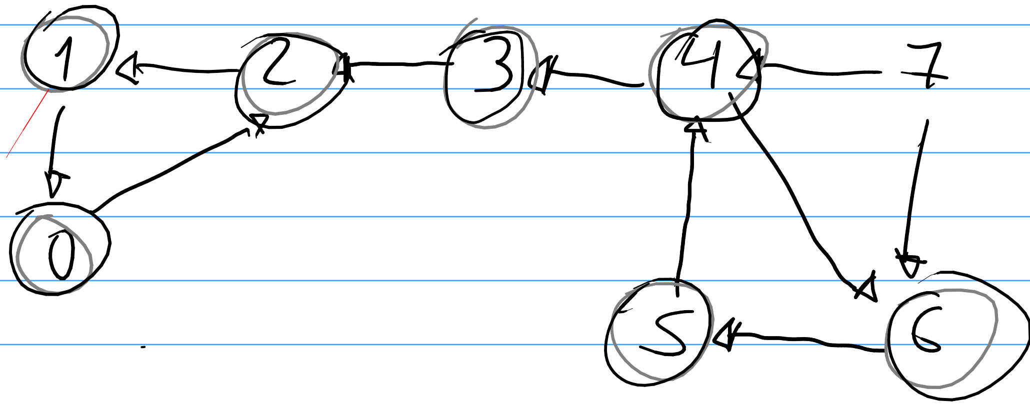
1. fazer uma DFS e guardar a ordem de finalização dos vertices em uma pilha
2. Inverter as direções do grafo original, obtendo o grafo transposto G^T
3. fazer uma DFS no grafo transposto, seguindo a ordem da pilha feita no passo 1.
4. Cada DFS identificará uma CFC.

ex =



Grafo transposto

Pilha = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7



CFCs = • 0, 1, 2 | • 7

• 3

• 4, 5, 6

Questão 3 (30 %)

Seja um grafo G com o seguinte conjunto de vértices $\{A, B, C, D, E, F\}$ e com a seguinte ponderação nas arestas.

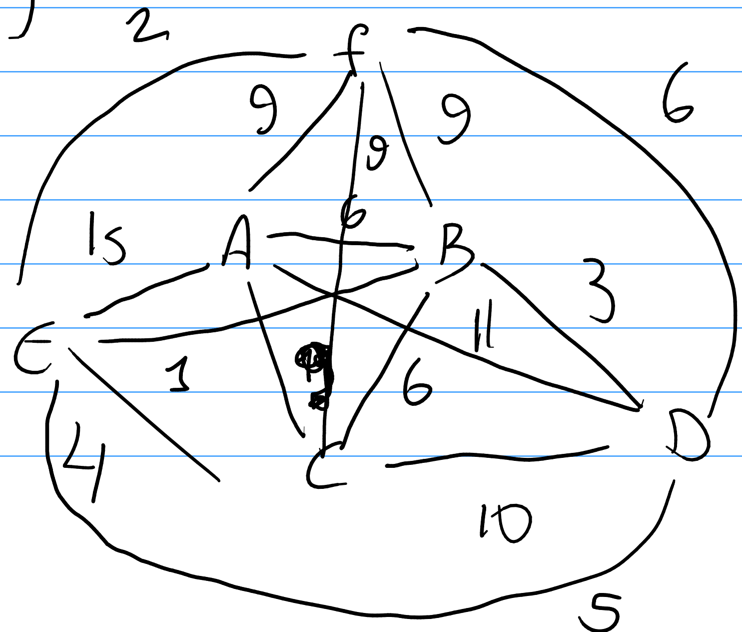
	A	B	C	D	E	F
A	0	6	9	11	15	9
B	6	0	13	3	1	2
C	9	13	0	10	4	4
D	11	3	10	0	5	6
E	15	1	4	5	0	2
F	9	2	4	6	2	0

a) (10 %) Discorra sobre o algoritmo de Kruskal, além disto, encontre a árvore geradora mínima de G usando o algoritmo de Kruskal (mostre o passo-a-passo de sua solução)

b) (10 %) Discorra sobre o algoritmo de Prim, além disto, encontre a árvore geradora mínima de G usando o algoritmo de Prim iniciando em F (mostre o passo-a-passo de sua solução)

c) (10 %) Discorra sobre o algoritmo de Dijkstra, além disto, encontre o menor caminho de F para todos os vértices em G usando o algoritmo de Dijkstra.

a)

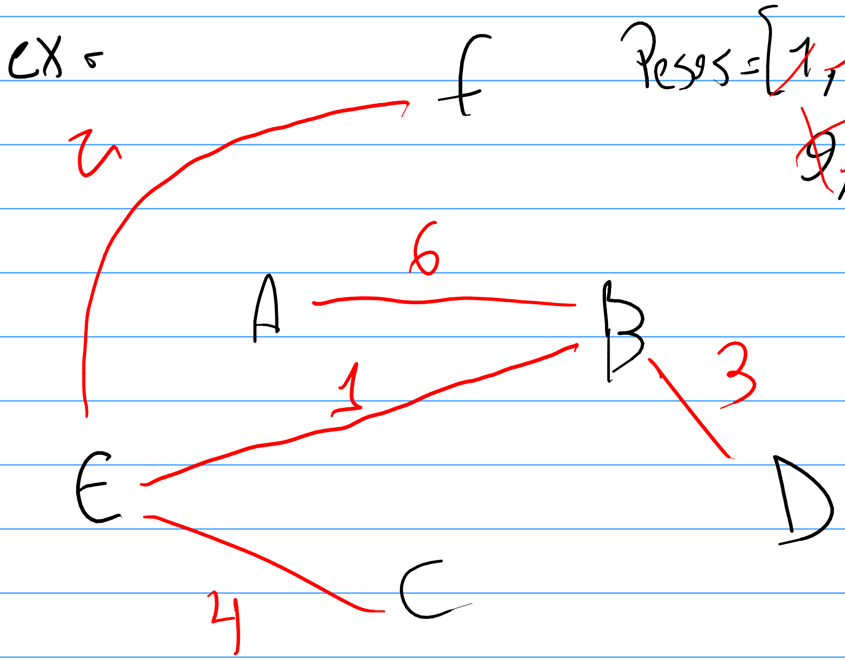


Kruskal é um algoritmo que nos dá a árvore geradora mínima de um grafo, isto é, um subgrafo conexo, acíclico e com a menor soma dos pesos;

1. Ordena os pesos em ordem crescente
2. Adiciona a aresta de menor peso na mst:
   - Se formar ciclo, pula para o próximo peso.
   - Senão, adiciona a mst
3. Processo até ter n-1 arestas, sendo n o número de vertices do grafo.

ex:

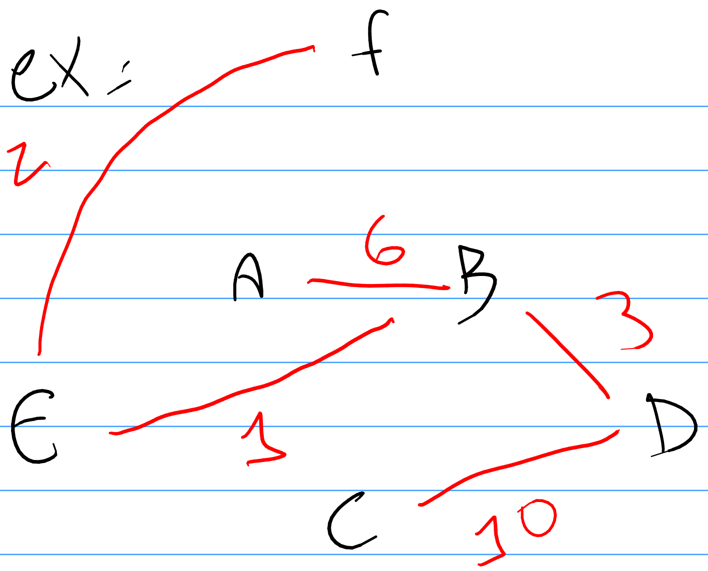
Pesos = [~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~6~~, ~~9~~, ~~9~~, ~~10~~, ~~11~~, ~~15~~]



b) O prim também é um algoritmo para se encontrar a MST:

1. Escolhe-se um vértice de origem
2. Para cada vizinho de v, adiciona na MST o com menor peso e que não forme ciclos
3. Faça isso até ter n-1 arestas.

ex:



início = F //

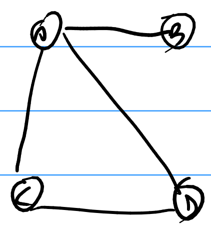
Questão 4

Discorra sobre subgrafo e subgrafo induzido. Apresente exemplos de ambos os conceitos.

Seja $G=(V,E)$ um grafo, um subgrafo $H=(V',E')$ é válido quando $H \subseteq G$, $V' \subseteq V$, $E' \subseteq E$.

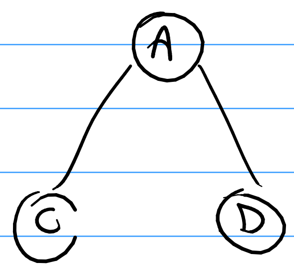
ex: $V=\{A,B,C,D\}$

$E=\{\{A,B\},\{A,D\},\{A,C\},\{C,D\}\}$



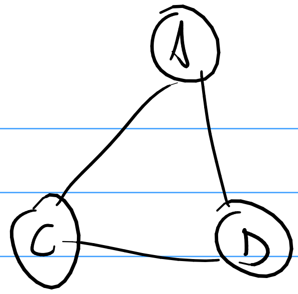
$V'=\{A,B,D\}$

$E'=\{\{A,C\},\{A,D\}\}$



Um subgrafo é dito induzido quando ao escolher o subconjunto V', o subconjunto E seja todos os arestas relacionados em V'.

ex: $V'=\{A,C,D\}$ | $E'=\{\{A,C\},\{A,D\},\{C,D\}\}$



Questão 1 - 35% Diâmetro em grafos (descreva em detalhes todas as soluções):

15 a) (15 %) Forneça um algoritmo (passo a passo) para calcular o diâmetro de um grafo não-direcionado. Apresente um exemplo que ilustre cada uma das etapas do método descrito.

20 b) (20 %) Seja G um grafo direcionado e fortemente conexo, forneça um algoritmo (passo a passo) para calcular o diâmetro em G. Apresente um exemplo que ilustre cada uma das etapas do método descrito. Discorra sobre o que ocorre se G não for fortemente conexo.

a) Caso o grafo seja conexo, o algoritmo será o seguinte:

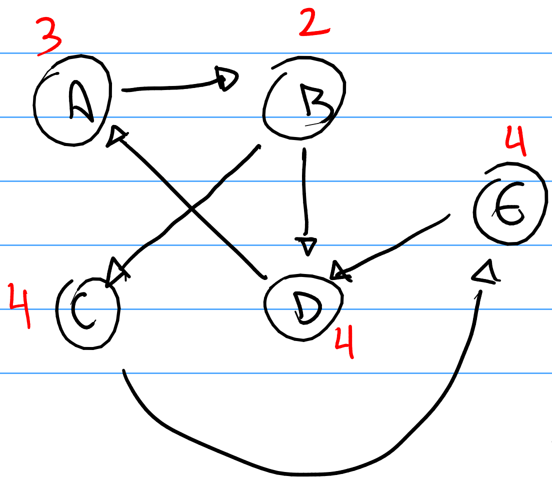
1. Faremos uma BFS a partir de um vértice qualquer v
2. Criamos uma lista de distâncias com o vértice de origem como $\emptyset$ e os demais com ∞, ao percorrer a busca, atualiza-se a distância. A excentricidade será a maior dessas distâncias, dada por: $\varepsilon(v) = \max_{u \in V}(d(v,u))$

3. Processamos todos os vértices, o diâmetro será a maior excentricidade entre todos os vértices, dado por: $\text{diâmetro} = \max_{v \in V}(\varepsilon(v))$

Caso o grafo não seja conexo, o diâmetro será ∞, pois haverá algum vértice isolado.

b) Seja $G=(V,E)$ um grafo direcionado e fortemente conexo, o mesmo algoritmo explicado anteriormente funcionaria.

ex =



$\varepsilon(A) = d[A:0,\ B:\overset{1}{\cancel{\infty}},\ C:\overset{3}{\cancel{\infty}},\ D:\overset{2}{\cancel{\infty}},\ E:\overset{3}{\cancel{\infty}}]$

$\varepsilon(B) = d[A:\overset{2}{\cancel{\infty}},\ B:0,\ C:\overset{1}{\cancel{\infty}},\ D:\overset{1}{\cancel{\infty}},\ E:\overset{2}{\cancel{\infty}}]$

$\varepsilon(C) = d[A:\overset{3}{\cancel{\infty}},\ B:\overset{4}{\cancel{\infty}},\ C:0,\ D:\overset{2}{\cancel{\infty}},\ E:\overset{1}{\cancel{\infty}}]$

$\varepsilon(D) = d[A:\overset{1}{\cancel{\infty}},\ B:\overset{2}{\cancel{\infty}},\ C:\overset{3}{\cancel{\infty}},\ D:0,\ E:\overset{4}{\cancel{\infty}}]$

diâmetro = 4

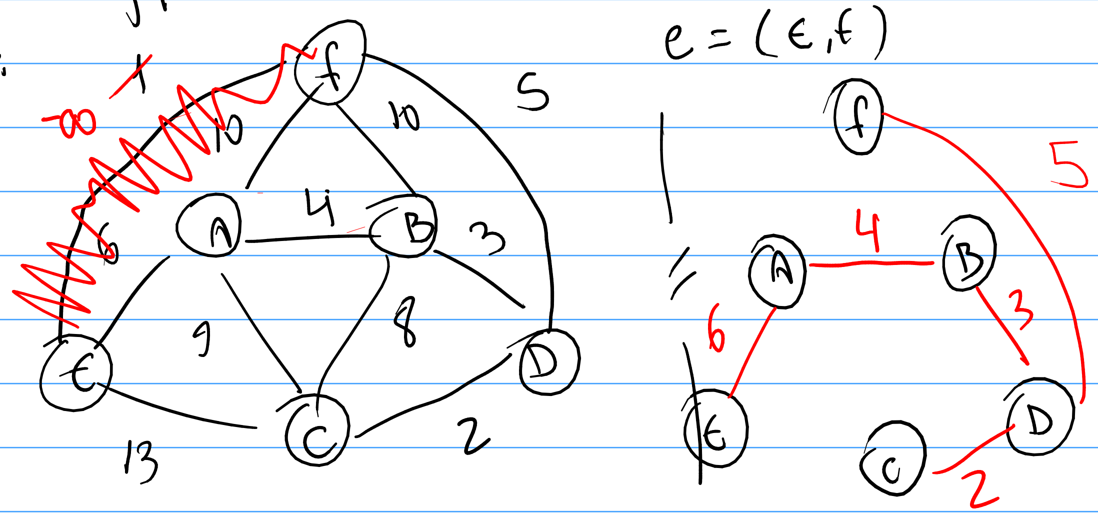
20

Questão 3 - 20% Seja um $G = (V, E)$ um grafo não-direcionado. Modifique o algoritmo do Prim para encontrar um árvore $T = (V, E')$ que seja um subgrafo de G de forma a garantir que uma aresta específica $e \in E$ (informada pelo usuário) sempre esteja presente em T e que $F = (V, E' \setminus \{e\})$ seja uma floresta geradora mínima de $G = (V, E \setminus \{e\})$. *Descreva em detalhes todas as soluções.*

1. Colocamos a aresta desejada com peso $-\infty$
2. Iniciamos de um vertice qualquer v e adicionamos na arvore as arestas vizinhas que possuem o menor peso e que não formam ciclos
3. Quando a árvore ter $|V|-1$ arestas, a MST está completa

Agora para garantir que na floresta não tenha a aresta e, removemos a aresta e executamos o prim normal, após a finalização adicionamos a aresta novamente ao grafo

ex:



10

Questão 4 - 10% É possível identificar se um grafo é acíclico usando o algoritmo para encontrar componentes fortemente conexos? Justifique sua resposta. *Descreva em detalhes todas as soluções.*

Sim, é possível identificar. Pois um componente fortemente conexo é um subconjunto de um grafo no qual há caminhos para quaisquer pares de vertices (u, v), tanto de u para v e tanto de v para u, ou seja, há um ciclo entre todos os vértices do componente. Por isso, ao identificar um CFC, se ele tiver mais de um vértice, é um ciclo.

Segue o algoritmo para encontrar CFC e ciclos:

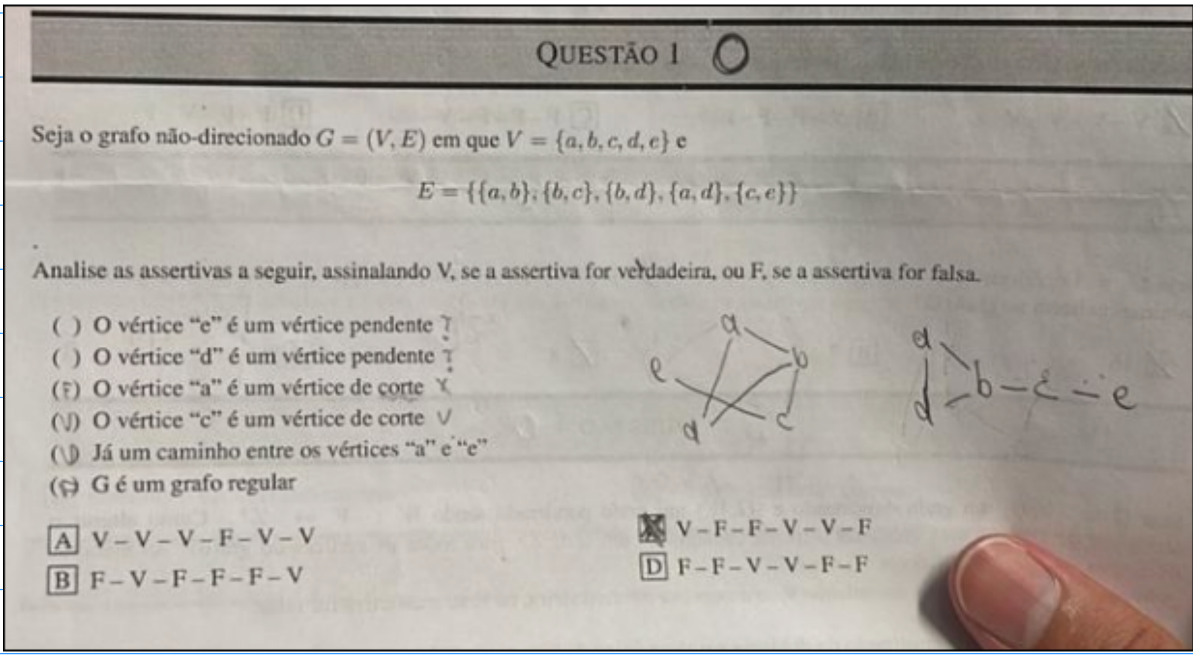
1. Inicializamos uma busca em profundidade (DFS) no grafo, guardando em uma pilha o tempo de finalização da busca para cada vértice.

2. Criamos um grafo transposto do grafo original, isto é, invertemos as direções do grafo.

3. Com o grafo transposto, fazemos outra DFS na ordem que os vértices estão na pilha feita na primeira busca.

4. Para cada DFS finalizada, temos uma CFC

5. Todos os CFC que tem mais que um vértice, é um ciclo.



QUESTÃO 1

Seja o grafo não-direcionado $G = (V, E)$ em que $V = \{a, b, c, d, e\}$ e

$$E = \{\{a,b\}, \{b,c\}, \{b,d\}, \{a,d\}, \{c,e\}\}$$

Analise as assertivas a seguir, assinalando V, se a assertiva for verdadeira, ou F, se a assertiva for falsa.

( ) O vértice "e" é um vértice pendente ?
( ) O vértice "d" é um vértice pendente ?
(F) O vértice "a" é um vértice de corte X
(V) O vértice "c" é um vértice de corte V
(V) Já um caminho entre os vértices "a" e "e"
(F) G é um grafo regular

A V – V – V – F – V – V

B F – V – F – F – F – V

C V – F – F – V – V – F

D F – F – V – V – F – F