Nome : Raylander Marques Melo

Matrícula: 494563

①

Suponha que exista um grafo conexo G, onde tenho uma aresta de peso mínimo (u, v), então para G ter uma árvore geradora mínima S, S tem que possuir um caminho que passe por todos os vértices e que a soma dos pesos das arestas seja a menor entre todos os outros. Assim seja S uma árvore geradora mínima do grafo G que não possui a aresta (u, v), assim suponha que exista uma árvore geradora mínima S do grafo que contém (u, v) desse modo chegamos em um absurdo, pois com S era mínima sem conter (u, v) e existe outra árvore mínima também S que contém (u, v), então o primeiro S não poderia ser mínima, assim percebemos que se a aresta (u, v) pertencer ao grafo ele estará contido a alguma árvore geradora mínima.

② Isso realmente acontece pois, uma aresta leve em qualquer corte feito em uma árvore geradora mínima qualquer T, desse modo, tem-se uma aresta (u, v) leve em T, um corte feito passando na aresta (u, v) que divide T em T_u e em T_v sendo elas componentes de uma floresta geradora mínima, assim a aresta (u, v) será uma aresta segura para realizar a ligação dessas componentes formando uma única componente sendo ela um árvore geradora mínima, pois se ela já fazia parte de uma árvore ela será uma aresta mínima do corte segura para refazer a ligação entre os componentes separados de T, então toda aresta leve contida em um corte que faz parte de alguma árvore geradora mínima, então ela é uma aresta leve que cruza um corte do grafo.

③

Suponha que existe um grafo G que possui uma aresta de peso máximo "e", então para possuir árvores geradoras mínimas que passem por todos os vértices do grafo tem-se duas possibilidades de geração de árvore de caminho mínimo:

1- O G tem que existir só uma árvore de caminho mínimo que utilize todos os vértices para que possa percorrer todos vértices, então só irá existir um caminho entre todo par de vértices, assim "e" será incluído na árvore de caminho mínimo pois só irá existir uma árvore de caminho mínimo.

2- O G tem que existir mais uma árvore de caminho mínimo que utilize todos os vértices, ou seja, G irá possuir ciclos, dessa forma se existe ciclos terá como passar por todos os vértices com mais de uma forma, ou seja, se "e" é a aresta de peso máximo ela irá na construção da árvore geradora mínima ela irá acabar ficando de fora por ser a aresta de maior peso e se existir um caminho menor que não passe por "e" ele será escolhido pelos critérios de criação de uma árvore geradora mínima.

Desse modo, no segundo caso temos que "e" não irá pertencer a pelo menos uma árvore geradora mínima, pois se não existir nenhuma árvore geradora mínima que "e" não pertença o primeiro caso é que será utilizado, e como este grafo possui pelo menos um ciclo, isso garante que existirá uma árvore geradora mínima que não possui "e".

(4)

Assumindo que existem duas árvores geradoras mínimas chamados T e T'. Para qualquer aresta "e" em T, se removermos uma aresta "e" pertencente a T, então T se desconecta e temos um corte (S, V-S). Então se um corte foi feito em uma árvore geradora mínima e na aresta "e" por onde o corte passa, "e" ela é leve através do corte (S, V-S) e segura para adicionar ela o árvore novamente pois ela já fazia parte da árvore geradora mínima. Se a aresta "se" está em T' e passa pelo corte (S, V-S), então "se" também é uma aresta de peso leve. Pois a aresta leve é somente uma, então "e" e "se" é a mesma aresta, e também está em T'. Dessa forma, sabemos que "e" foi escolhido de forma aleatória, de todas as arestas de T, ou seja, isso também aconteceu em T'. Assim temos a conclusão de que a árvore geradora mínima é uma só.

⑤

O motivo pelo qual pode acontecer de se ter várias árvores geradoras mínimas diferentes é que temos várias opções de arestas com o mesmo peso no grafo.
Dessa forma, dada uma árvore geradora mínima T, desejamos ordenar as arestas no algoritmo de Kruskal de modo que produza T. Para que isso aconteça temos que cada aresta "e" em T e o grafo não tenham valores de arestas iguais e possua somente uma estrutura que gera T como árvore geradora mínima, ou seja, para todo corte feito em T terá apenas uma aresta leve, assim faz com que só se possa conseguir um caminho mínimo de uma aresta para outra, ou seja, dessa forma não tem como o algoritmo de Kruskal não construir T com peso $w(e)$, pois só existirá esta árvore geradora mínima no grafo.

⑥

a) Este algoritmo conseguiu construir uma árvore geradora mínima, pois o algoritmo começa ordenando as arestas de forma não crescente dos pesos, após isso ele adiciona todas as arestas a um grafo T, quando todas as arestas estão adicionadas é feito uma busca em todas as arestas e verifica se aquela aresta pegada de acordo com a ordenação das arestas feitas acima for retirada de T, mantém T conexo, e após esta busca a árvore fica completa, ou seja, esse algoritmo gera uma árvore geradora mínima de qualquer grafo conexo, pois a retirar todas as arestas em ordem não crescente e que não desconecta T, o algoritmo está tirando uma aresta muito pesada de forma que não faz parte daquela árvore geradora mínima, assim quando não houver mais arestas que possam ser retiradas, todas as arestas mais pesadas já foram retiradas e a árvore geradora mínima está pronta.

b) Este algoritmo não conseguiu construir uma árvore geradora mínima, pois o algoritmo começa deixando a árvore geradora mínima vazia, depois é feito uma varredura por todas as arestas do grafo que quando adicionar qualquer aresta aleatória do grafo

se ela não gerar um ciclo esta aresta é adicionada a árvore geradora mínima, dessa forma o algoritmo não garante que a aresta que está sendo adicionada no grafo é uma aresta mínima, assim esse algoritmo não gera árvore geradora mínima.

C) Este algoritmo consegue construir uma árvore geradora mínima, pois o algoritmo começa criando a árvore geradora mínima vazia, depois é feito uma varredura por todas as arestas do grafo que pega qualquer aresta e adiciona na árvore geradora mínima e depois é feito uma verificação dentro do mesmo passo da varredura que se tiver um ciclo na árvore geradora mínima ele vai retirar a aresta de peso máximo no ciclo, assim quando o algoritmo terminar de executar ele vai garantir a árvore pois vai ter sido retirado todas as arestas mais pesadas e foi garantido que não existe ciclos, assim o algoritmo funciona.

(7)

A opção correta é a C).

O algoritmo de Bellman-Ford ele irá percorrer todos os vértices de um grafo ponderado com pesos procurando qual o melhor caminho para os vértices onde ele possui uma aresta que se conecta formando um arco até aquele vértice, ou seja, após o algoritmo percorrer todos os vértices ele irá ter pego o melhor caminho, por ter visto qual a melhor possibilidade entre cada par de vértice, detalhe, este algoritmo verifica se existe um ciclo negativo e diz se tem solução para aquele grafo ou não, se tiver solução ele devolve a solução e se existir um ciclo negativo ele irá dizer que aquele grafo não existe soluções.

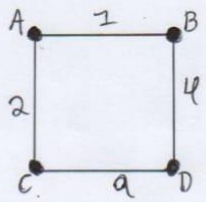
Já o algoritmo de Dijkstra ele percorre o grafo vértice a vértice verificando se a melhor ligação para as arestas adjacentes a que ele acabou de descobrir, assim relaxando todas as arestas, sabendo que este algoritmo não é capaz de identificar ciclo negativo, ou seja, essas arestas não podem possuir valores negativos.

Já o algoritmo de Kruskal ele é um algoritmo guloso utilizado em árvores não direcionadas, ou seja, o algoritmo de Kruskal ele vai vendo os pesos das arestas e vai adicionando de forma não decrescente, ou seja, ele pega a menor aresta se tiver duas iguais ele vai lá e verifica se aquela aresta não fecha um ciclo se ela não fecha ela é adicionada a árvore geradora mínima, ou seja, quando ele verificar todas as arestas irá possuir uma árvore geradora mínima.

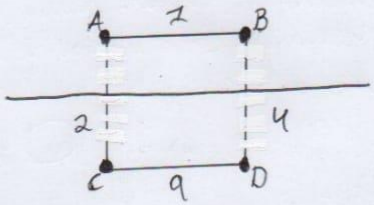
Já o algoritmo de Prim tem um pouco de semelhança com o de Kruskal por ser um algoritmo guloso, porém o de Prim irá pegar um vértice por onde ele vai começar a fazer o caminho, e a partir desse vértice ele irá vê qual a aresta que liga esse vértice a outro tem o menor peso e adiciona ele ao caminho da árvore geradora mínima e depois passar para o vértice que foi feito a ligação e repetir a verificação de qual é a aresta de menor peso e assim sucessivamente até passar por todo o grafo e assim conseguir construir a árvore geradora mínima.

Agora o algoritmo que é a opção correta da questão, o algoritmo de Floyd - Warshall ele literalmente vai vê todas as possibilidades do grafo direcionado com pesos, ou seja, este algoritmo irá verificar qual o menor caminho entre todos os pares de vértices, ou seja, ele seleciona um vértice e verifica qual o melhor caminho daquele vértice até todos os outros vértices do grafo e isso é realizado para cada vértice do grafo, dessa forma esse algoritmo garante que todas as possibilidades seja testadas. Observação este grafo não pode possuir ciclo negativo, pois caso exista ele vai resultar em respostas inválidas.

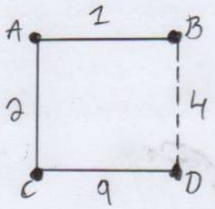
(B) O algoritmo falha, vamos ver o exemplo abaixo:



Crie um grafo G que vai particionar de V_1 e V_2 da forma que $V_1 = \{A, B\}$ e $V_2 = \{C, D\}$ e as arestas $E_1 = \{(A,B)\}$ e $E_2 = \{(C,D)\}$ como mostrado abaixo:



Assim para criar uma árvore geradora mínima de acordo com o algoritmo temos que adicionar a aresta leve do corte que neste caso é a $E = \{(A,C)\}$ assim quando adicionamos ela a árvore geradora mínima fica como abaixo:



Essa não é a árvore com menor peso, então o algoritmo não funciona para todo grafo.

④ Se $v \in V$ pode ser atingido por s, então existe s ~~e~~ u → v um caminho mínimo de s a v em G. Suponha que G foi inicializado a partir do algoritmo de inicialização e depois uma sequência de etapas de relaxamento é executado para todas as arestas de, então após o relaxamento da aresta (u,v), tem-se:

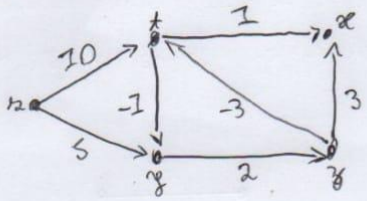
$Dv \leq Du + W((u,v))$

$\delta(s,u) + W((u,v))$

$Dv \geq \delta(s,v)$

Da qual concluímos que essa igualdade é mantida daí em diante, ou seja, quando termina a execução de Bellman-Ford é concluída $v.d \leq \infty$.

(10)



O motivo desse algoritmo dá errado é que ao haver um ciclo negativo como no exemplo, quando o algoritmo é executado ele irá mostrar os valores de custo para chegar em t, y, x e z errados pois ele passou pelo ciclo negativo no máximo uma vez, então essa não será a árvore mínima, pois basta percorrer o ciclo novamente que vai achar um valor menor do que já foram encontrados.

(II) Isso ocorre pois se temos que G é um grafo ponderado com s como fonte, s não pode possuir pai, ou seja, s não pode ser descendente de ninguém, pois ele faz o início do percurso, então se o pai de s não for nulo isso significa que s é descendente de alguém, e como temos um grafo ponderado e só pode ter um de s para qualquer outro vértice do grafo ~~sem~~ sem volta para não ocorrer ciclo, assim com s.d sendo diferente de nulo temos que o grafo possui um ciclo e se essa aresta que fecha um ciclo faz parte da árvore ~~mesmo~~ mesmo após o relaxamento ela tem que ser uma aresta leve, ou seja, se o caminho de s para todos os vértices já tinha sido construído com a arestas mais leves se outra aresta é adicionada para chegar em s ela tem que ser negativa, pois se não ela não seria adicionada, desse modo, gerando o ciclo negativo.