



UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Estructuras Discretas II

Docente: Carlo Corrales Delgado

Ejercicios de Lección 5 C

Escuela:

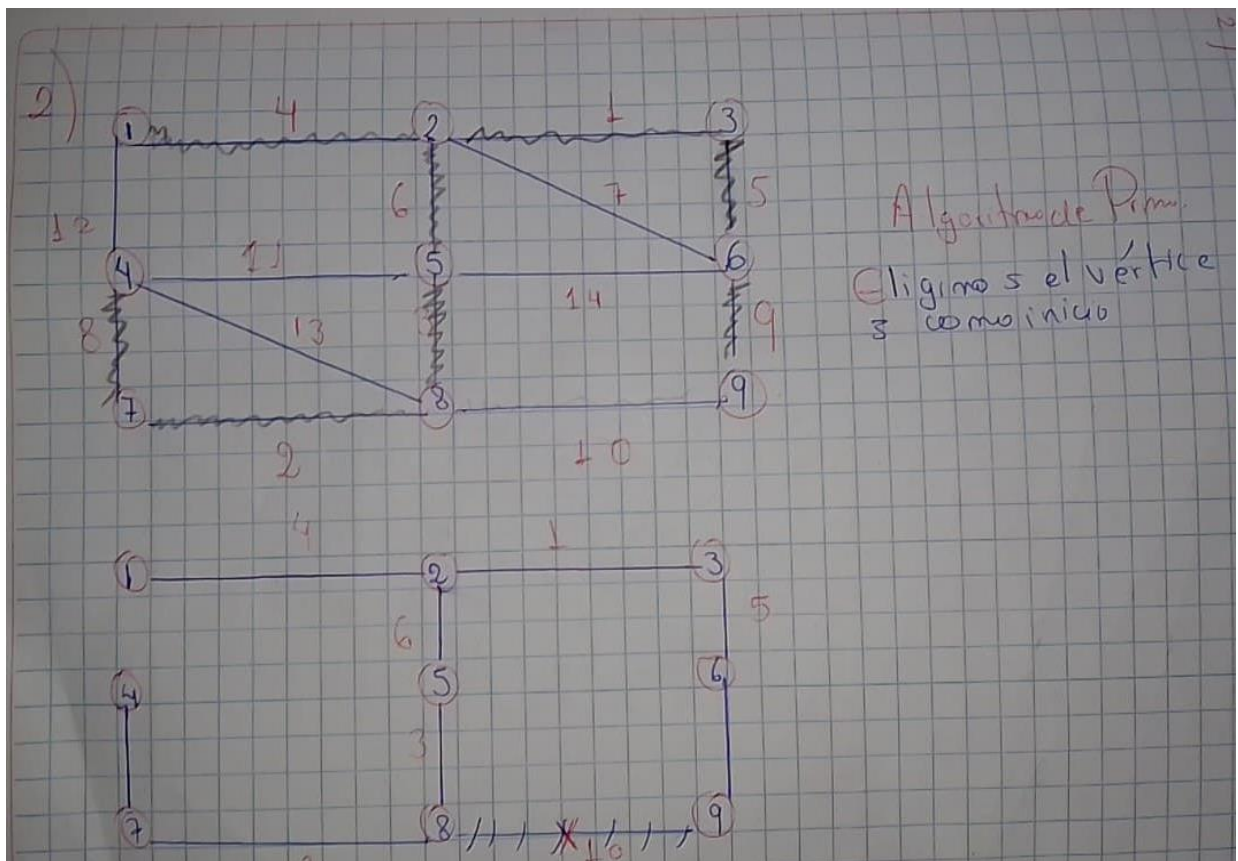
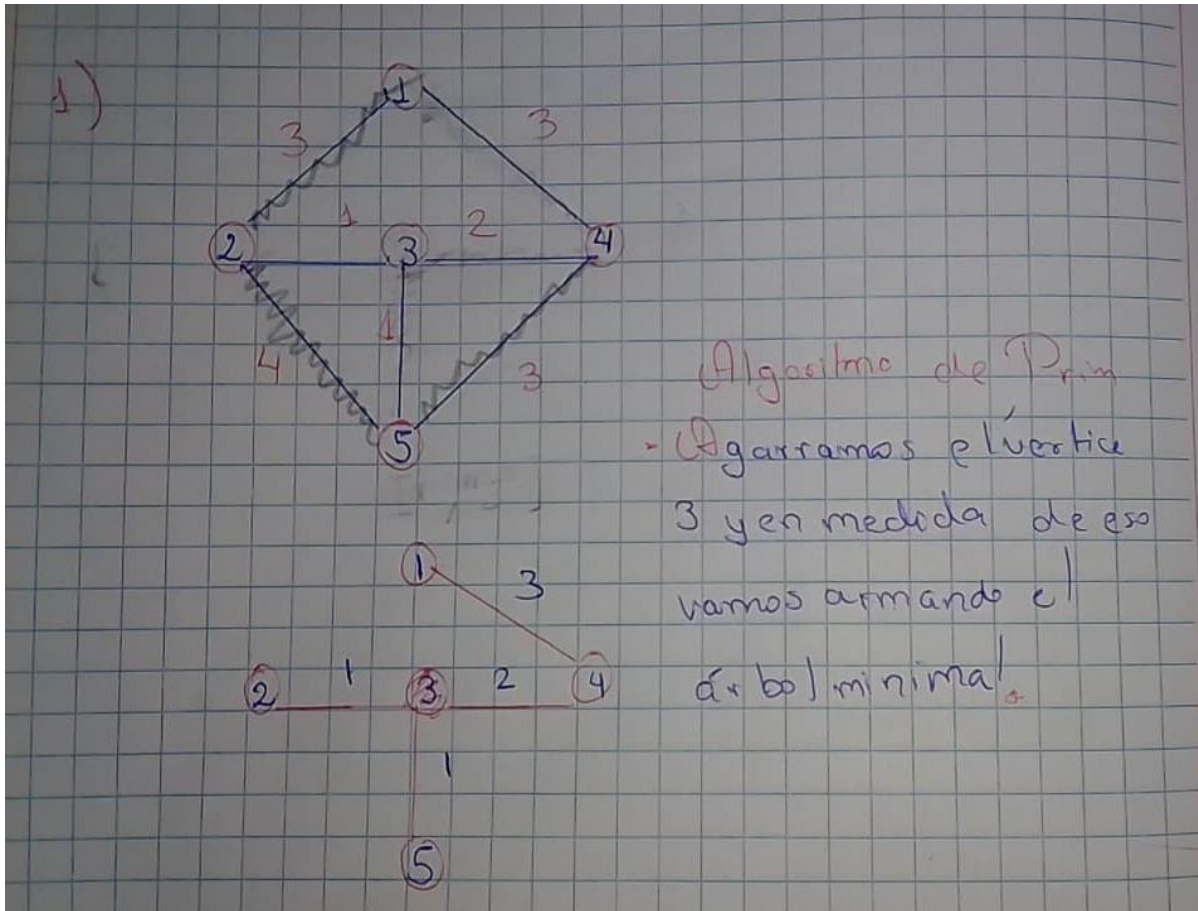
Ciencia de la computación (Primer año)

Temas:

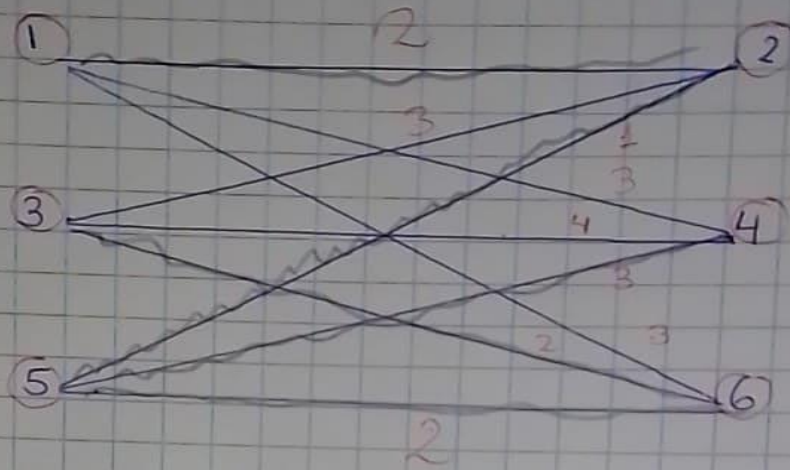
-Prim y Kruskal

Alumno:

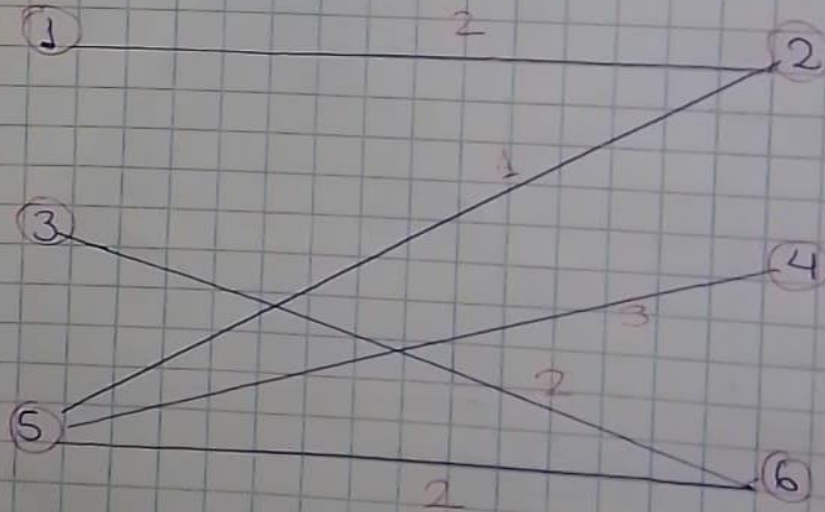
Josue Gabriel Sumare Uscca



3)



Algoritmo de Prim

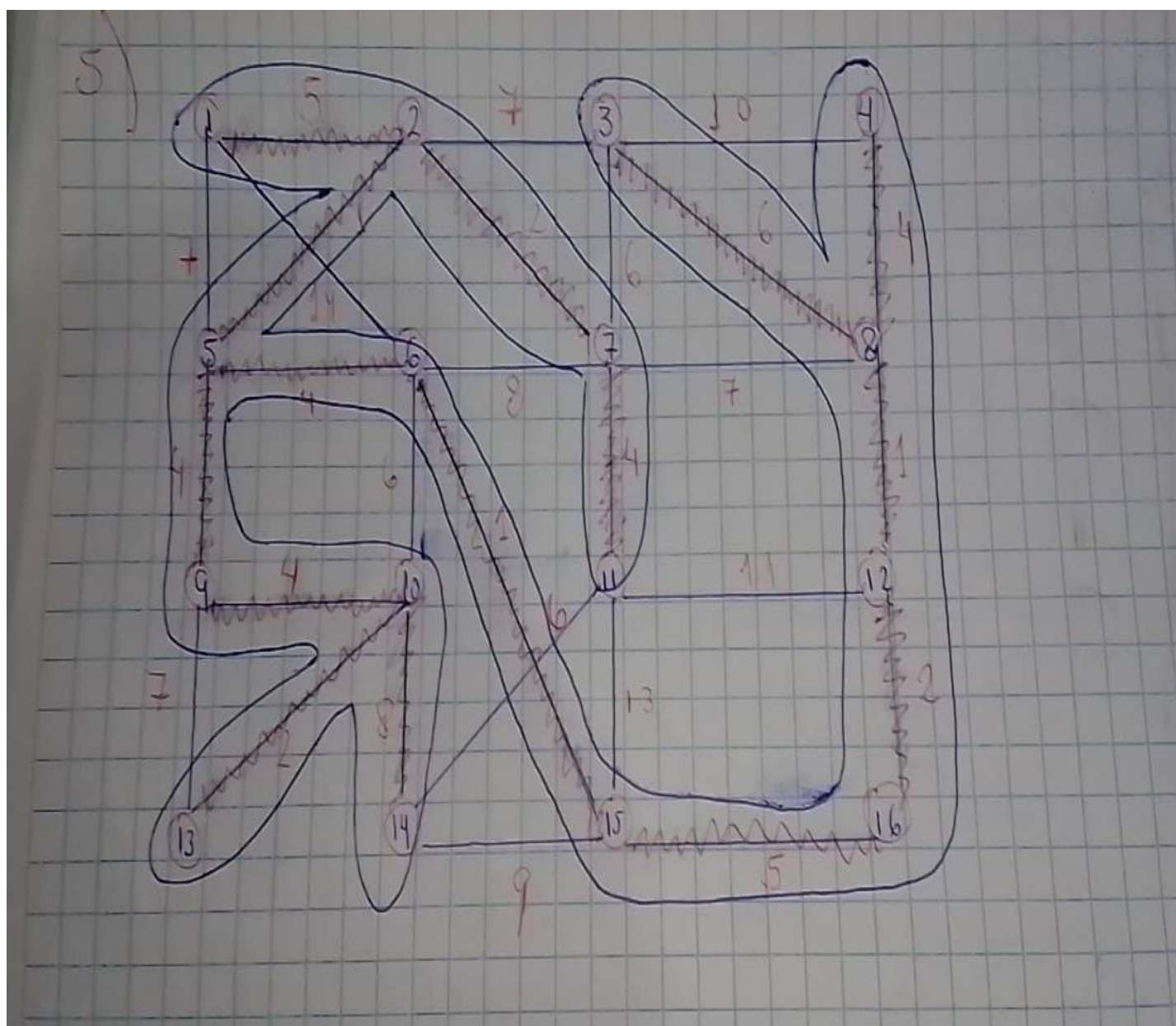


A hand-drawn graph on grid paper showing 12 nodes arranged in a 4x3 grid. Nodes are numbered 1 through 12. Edges connect adjacent nodes horizontally, vertically, and diagonally. Each edge has a weight value written next to it. The weights are:

- (1,2) = 3
- (2,3) = 4
- (3,6) = 6
- (6,9) = 6
- (9,12) = 7
- (12,11) = 2
- (11,8) = 5
- (8,5) = 2
- (5,4) = 3
- (4,1) = 10
- (1,4) = 4
- (4,7) = 6
- (7,10) = 3
- (10,11) = 3
- (11,8) = 5
- (8,5) = 2
- (5,4) = 3
- (4,1) = 10
- (1,4) = 4
- (4,7) = 6
- (7,10) = 3
- (10,11) = 3
- (11,8) = 5
- (8,5) = 2
- (5,4) = 3
- (4,1) = 10

A weighted undirected graph with 12 vertices labeled 1 through 12. The vertices are arranged in a grid-like structure. The edges and their weights are:

- (1,2) weight 3
- (2,3) weight 4
- (1,5) weight 2
- (5,6) weight 3
- (4,7) weight 4
- (7,10) weight 3
- (5,8) weight 2
- (8,11) weight 5
- (8,10) weight 3
- (11,12) weight 2
- (9,12)

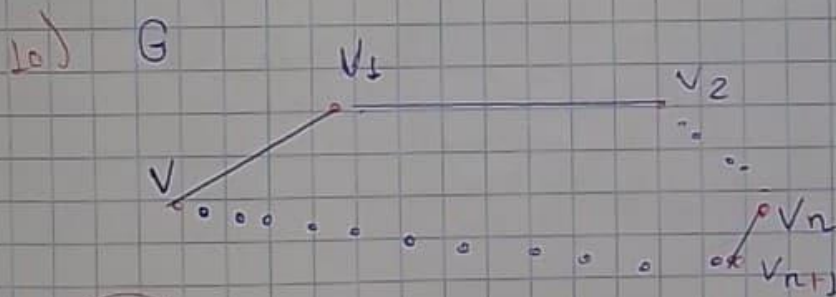


6) Al haber tres bucles for si se cumplen que se pueden agregar pues pasan por los 3 bucles volviéndose un n^3

7) Se aplica lo mismo que el Algoritmo de Prim; agregándole el concepto de etiquetas temporales y permanentes, conceptos presentes en el algoritmo de Dijkstra.

8) En el peor caso entra no solo en el bucle for sino de cumplirse la condición del while la complejidad entrara a n^2 .

9) Ya que las aristas que se eligen son las aristas con menor peso has q llegar a conectar a todos con el mínimo de conexiones con las aristas de menor peso.



Para cualquier algoritmo siempre se toma a la arista con menor peso sin importar si en el proceso esta componente esta aislado al grafico

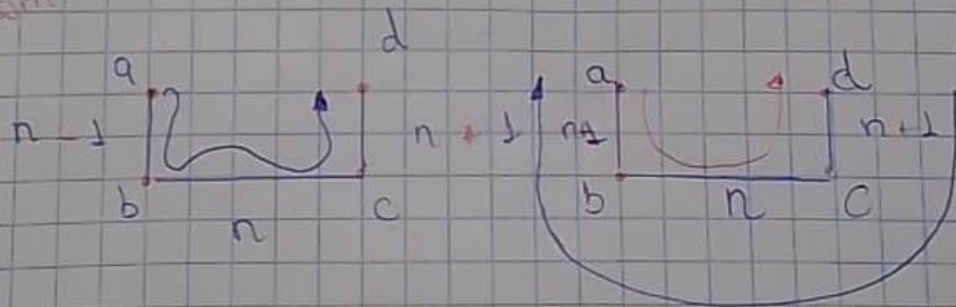
11) Si debido a que todo algoritmo de un árbol minimal se empieza por las aristas mínimas, o se borran las aristas máximas por lo que el más mínimo no se borrará nunca.

12) Si aplicamos el algoritmo de Prim; Kruskal se evalúa los mínimos o máximos, sin embargo usar esto será inútil debido a que todos tienen el mismo peso por lo que el árbol minimal se tendría que evaluar en todas las aristas con mismo peso.

13) Al ser conexa y diferentes en peso es más factible aplicar un algoritmo generador de árbol minimal.

14) No, ya que al ser diferentes todas las aristas el árbol minimal será uno solo, ya que no pueden haber vertices iguales entonces no hay aristas mínimas de igual peso.

Sim



15) Si, debido a que este al no forma un ciclo, porque como sabemos primero agregamos las aristas menores, por lo que este no formaría un ciclo y siempre estaría en el árbol mínimo.

16) No debido a que en gráficos como el K_5 las etiquetas no servirían, debido a que las etiquetas no cambiarían y el concepto de etiquetas no sería útil.

17) Si es posible, debido a que de este paso es parte del algoritmo de Kruskal en el cual se eliminaba las aristas de mayor peso.

18) algoritmo Arbol Mínimo (listaDeAristas, listaDePesos)

// Hacemos $\text{for } \text{int } i = 0, i < \text{longitud}(\text{listaDeAristas}), i++$
// recorriendo
// por cada la
// arista

// otro recorriendo $\text{for } \text{int } j = 1, j < \text{longitud}(\text{listaDePesos}), j++$
// para que compare

en j para con i
para ya i con un i

// Condición de $\text{if } (\text{listaDePesos}[i] > \text{listaDePesos}[i-1])$

ser mayor como
en el anterior como
tanto lo explicamos

// Esto para parar el
borrado de los mayores
en pesos.

$\text{if } (\text{listaDePesos} = \text{long}(\text{listaDeAristas}) - 1)$

return listaDePesos

// Borra las aristas con
mayor peso

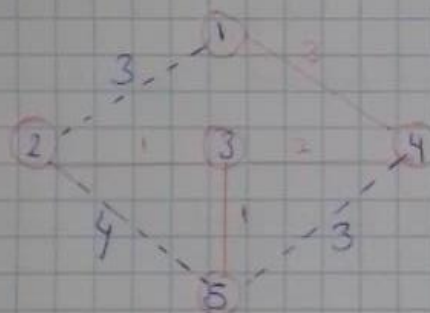
else { // al borrar no debe formarse un
borrar(i) de listaDePesos
borrar i perteneciente a listaDeAristas

19) Funciona, debido a que es un algoritmo basado en el algoritmo de Kruskal, cuyo algoritmo es comprobado.

20) Es un algoritmo con pasos totalmente contrarios al algoritmo de Prim, que nos lleva a ingresar primero las aristas mínimas, de lo contrario el algoritmo de Kruskal es eliminar los valores más grandes en forma progresiva sin eliminar aquellos que formen más de una componente.

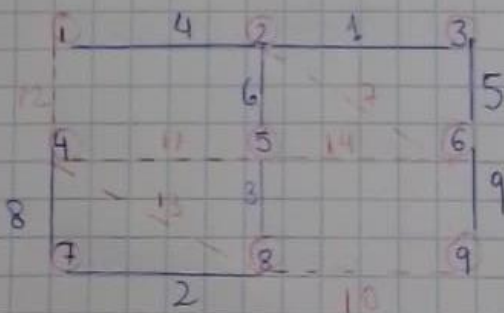
21)

①



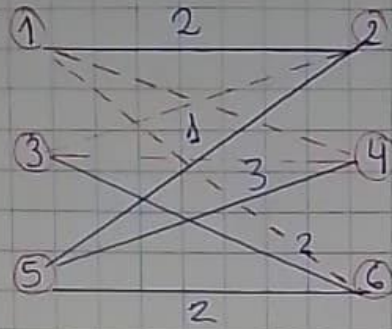
Se elimina por ser aristas con pesos máximos.

②



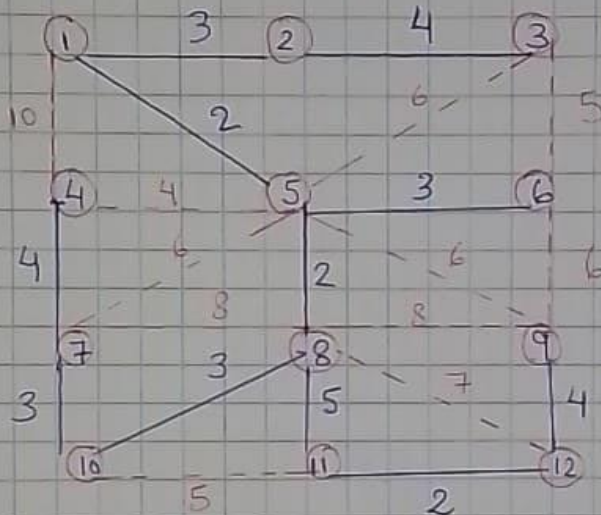
Se elimina por ser aristas con peso máximo.

3)



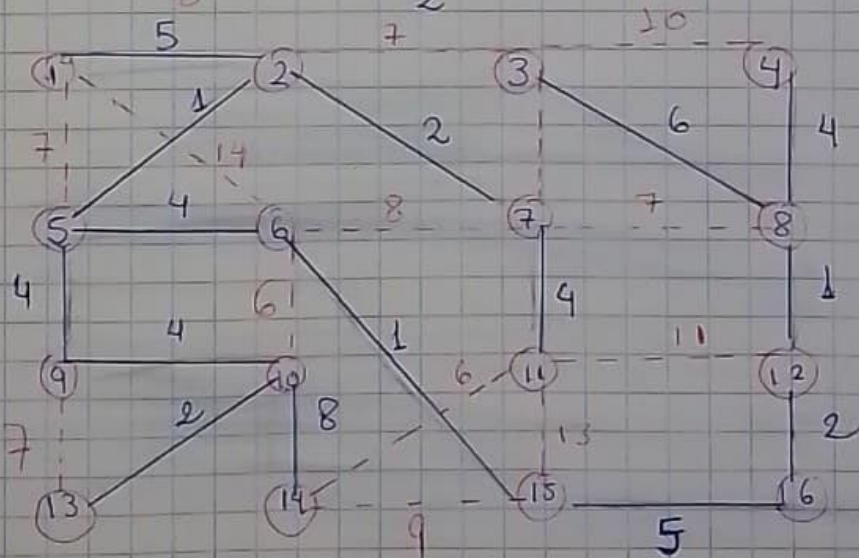
--- : Se eliminan
por tener
máximos
pesos =

4)



--- : Se eliminan
las aristas
con mayores
pesos, sin
formar más
de una compo-
nente.

5)



22) A) eliminar las aristas de mayor peso
Solo quedan las aristas de menor peso que
conecten todos los vértices.