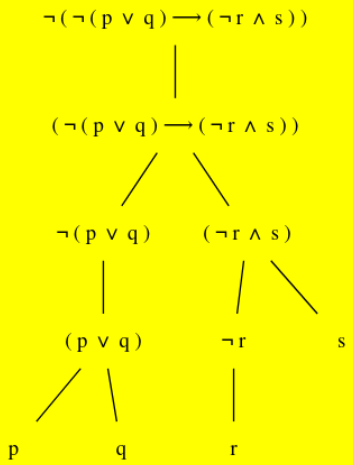
★ Sea G=(V,A) una gráfica, no necesariamente conexa, y un entero positivo k.
Sea E una expresión lógica en su forma normal conjuntiva,
PC: Determinar si G tiene un clan de orden mayor o igual que k.
Γ : Determinar si G tiene una subgráfica conexa con al menos k vértices.
Ψ: Determinar si G una 4-coloración válida.
Φ : Determinar si E se satisface, donde cada cláusula tiene 4 variables,

1. Proporcionar explícitamente un algoritmo No-determínistico Polinomial para los **dos** de los problemas Γ, Φ y Ψ correspondientes.
   Primero daremos la solución de Ψ

   a) Se deberá especificar explícitamente **cómo** usa la primitiva elección-nd;
   Respuesta
   Tomaremos la expresión lógica y las dividiremos por secciones, con esto nos referimos a que cada variable tendrá un color, por ejemplo proponemos esta expresión lógica : $\neg(\neg(p \vee q) \wedge (\neg r \vee s))$

   En este caso A tendrá el color rojo, B azul, D rosa y E amarillo.

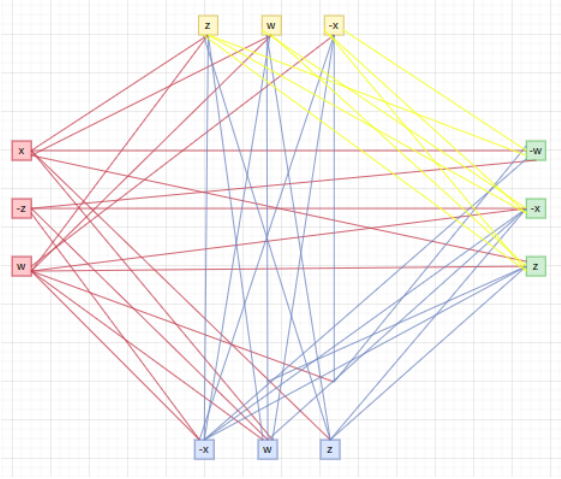
   b)Se deberá proporcionar explícitamente el algoritmo de verificación.
   Respuesta
   Verificamos si es correcta con la equivalencia semántica de la expresión lógica dada.



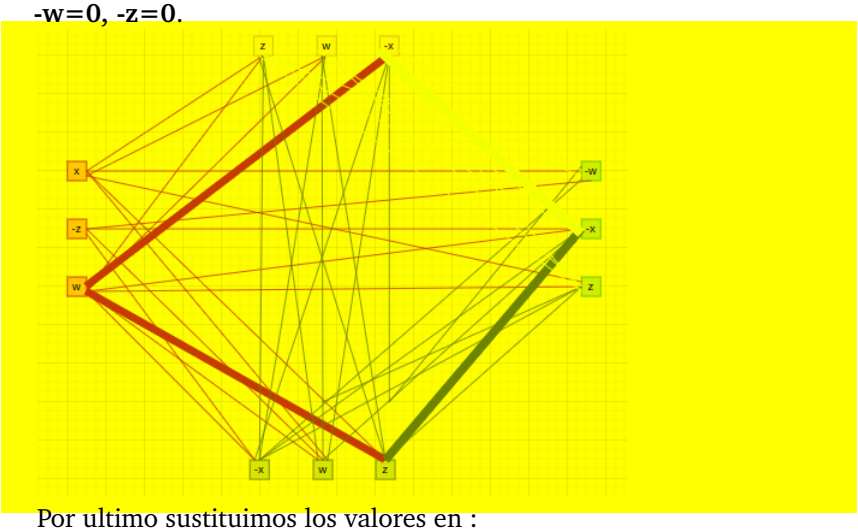
   Ahora contestaremos Φ

   a) Se deberá especificar explícitamente **cómo** usa la primitiva elección-nd;
   Respuesta
   Para este problema lo primero que debemos de hacer es recibir una expresión lógica y después asignarles valores verdaderos (Verdadero o Falso) a la expresión lógica, estos valores se asignan de manera al azar.
   Por ultimo tenemos que verificar si se satisface la expresión lógica
   b)Se deberá proporcionar explícitamente el algoritmo de verificación.
   Respuesta
   Después de verificar la expresión lógica, si este es valido, significa que las clausulas de la expresión se cumplieron y si no, significa que no es satisfacible esa expresión lógica.

2. Para el problema **PC** y la Expresión lógica E construir la gráfica G e $ilustrar$ la **Afirmación A**. Debes proporcionar la asignación de verdad para E y mostrar explícitamente el clan en la gráfica G... **y viceversa**.
   Afirmación A: E se satisface si y sólo si G tiene un clan de tamaño m.
   $E_A = (x + z + w) * (x + w + z) * (w + x + z) * (z + w + x)$;
   v es la negación de v
   Respuesta
   Primero mostraremos la gráfica G que se nos da:

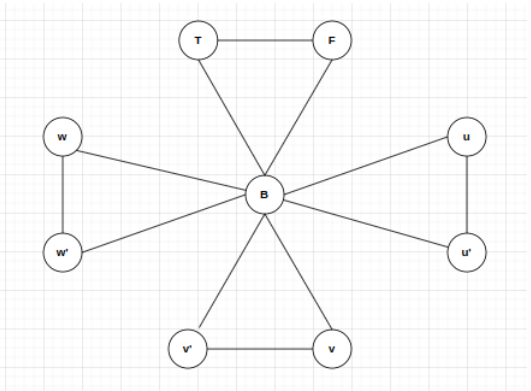


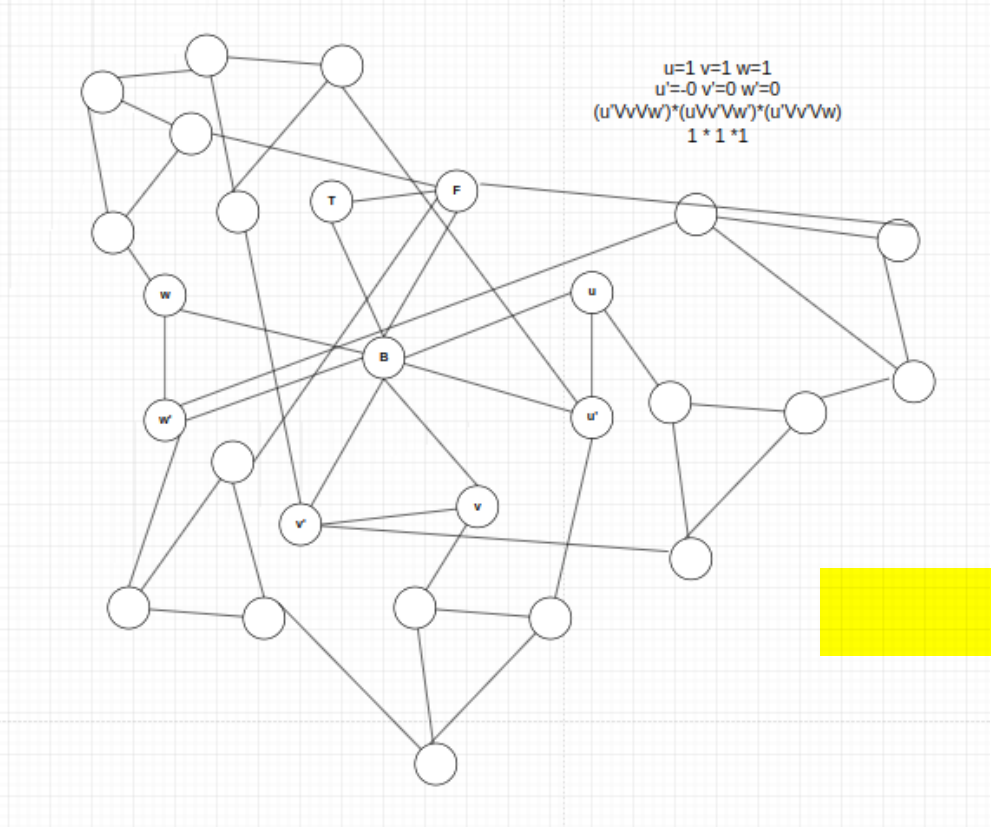
Y encontramos el clan de **-x w z -x - x**, donde **-x=1, w= 1 y z= 1** y por ende sus valores son **x=0, -w=0, -z=0**.



Por ultimo sustituimos los valores en :
$E_A = (x + z + w) * (x + w + z) * (w + x + z) * (z + w + x)$; y nos queda:
$E_A = (0 + 0 + 1) * (-1 + 1 + 1) * (0 + 0 + 1) * (1 + 1 + -1)$;
$E_A = (1) * (1) * (1) * (1) = 1$ y con esto podemos decir que si se cumple, es correcta su validez y se cumple la condición de E_A

3. Para el problema 3-Coloración considera la reducción presentada en clase, con 3- SAT, donde se da la **Afirmación C.** Ilustra tal afirmación con una Expresión lógica de al menos tres cláusulas.
Afirmación C: Una gráfica G es 3-coloreable si y sólo si E se satisface.
Respuesta
Sea G $f = (u' \vee v \vee w') * (u \vee v \vee w') * (u' \vee v' \vee w)$

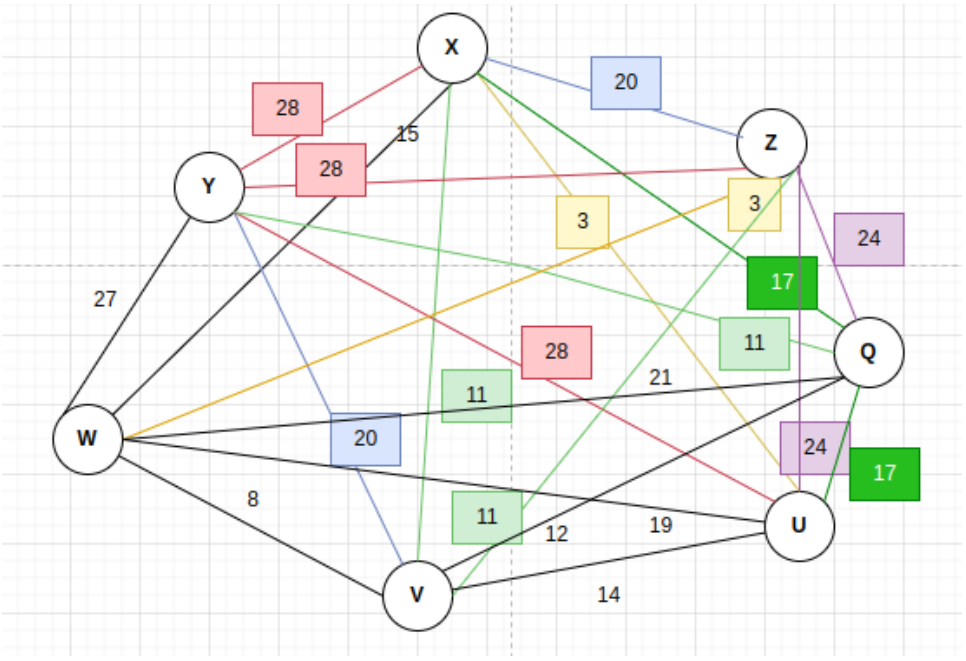




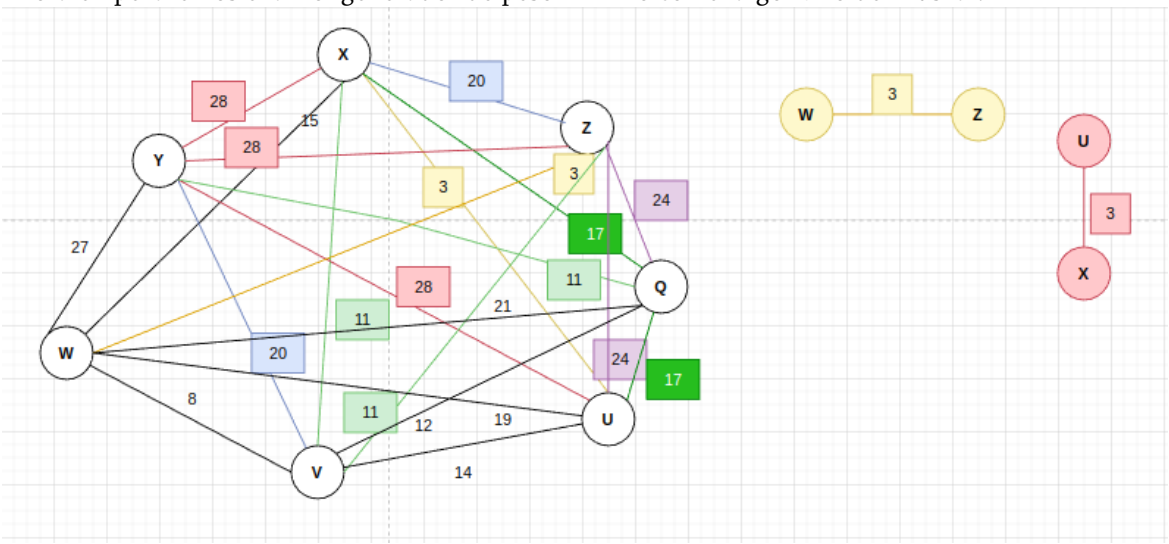
4. Para la siguiente matriz de adyacencias, un ejemplar del Agente Viajero (TSP) aplicar el primer algoritmo de aproximación dado en las notas y generar tour. Deberás detallar cada paso del algoritmo e ilustrar cómo se tomaron los atajos.

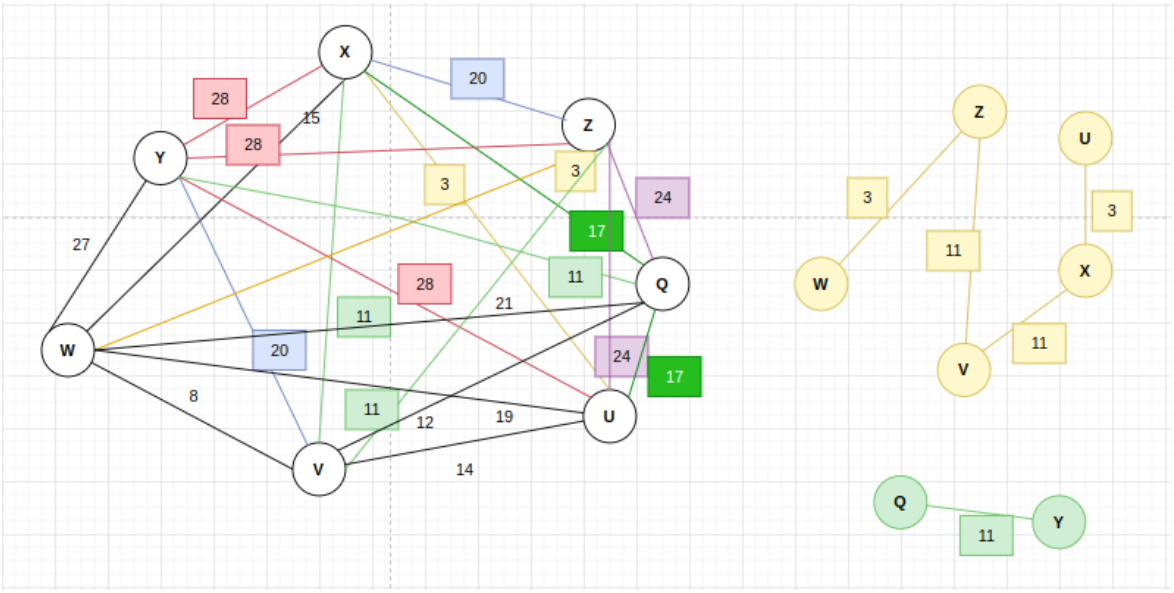
c	X	Y	Z	W	V	U	Q
X	0	28	20	15	11	3	17
Y	28	0	28	27	20	28	11
Z	20	28	0	3	11	24	24
W	15	27	3	0	8	19	21
V	11	20	11	8	0	14	12
U	3	28	24	19	14	0	17
Q	17	11	24	21	12	17	0

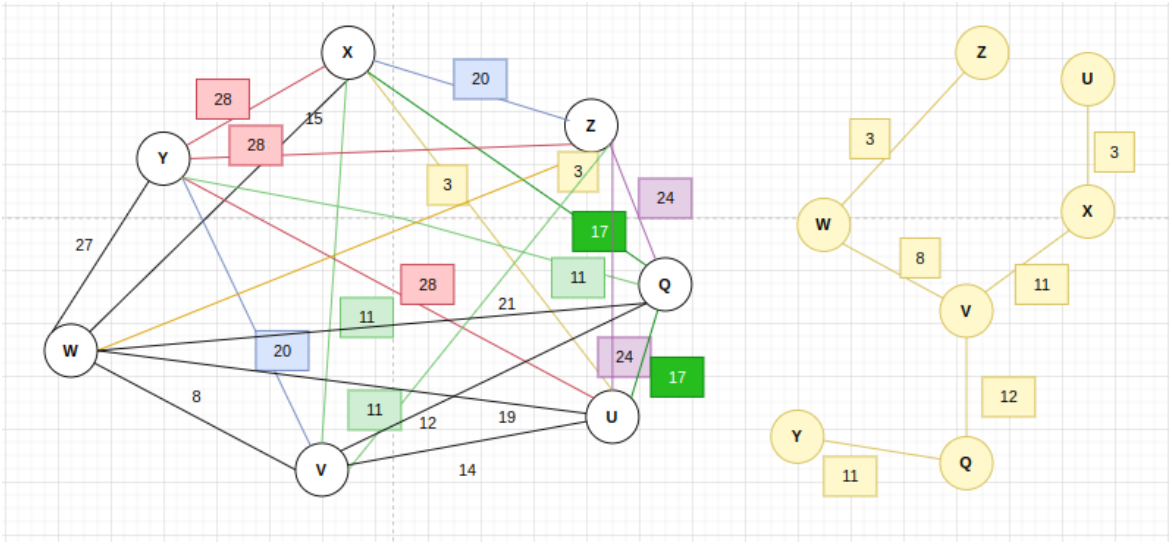
Respuesta
Con la información de la tabla obtenemos esta gráfica

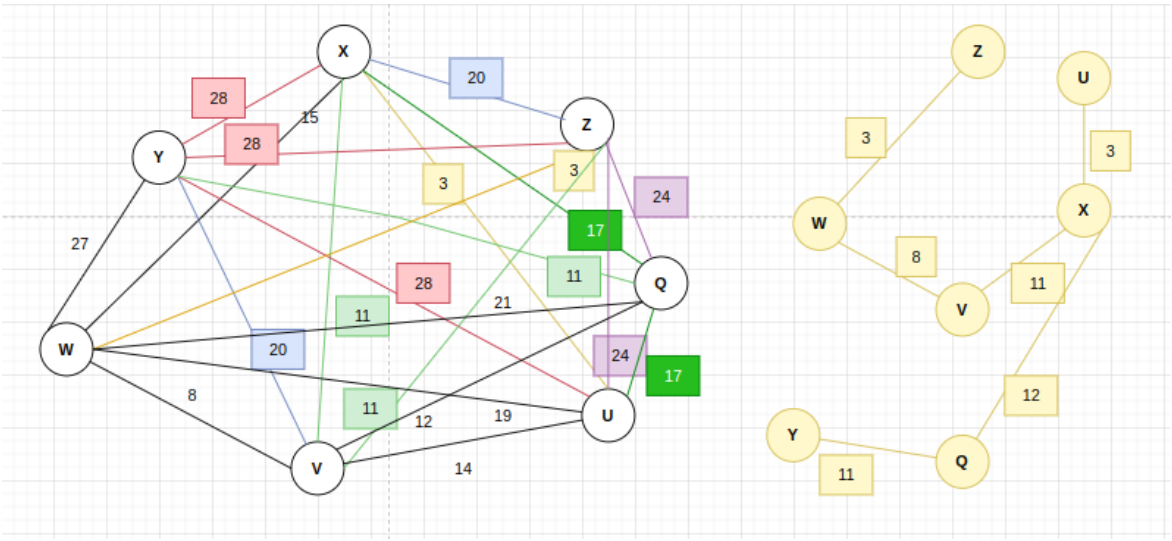


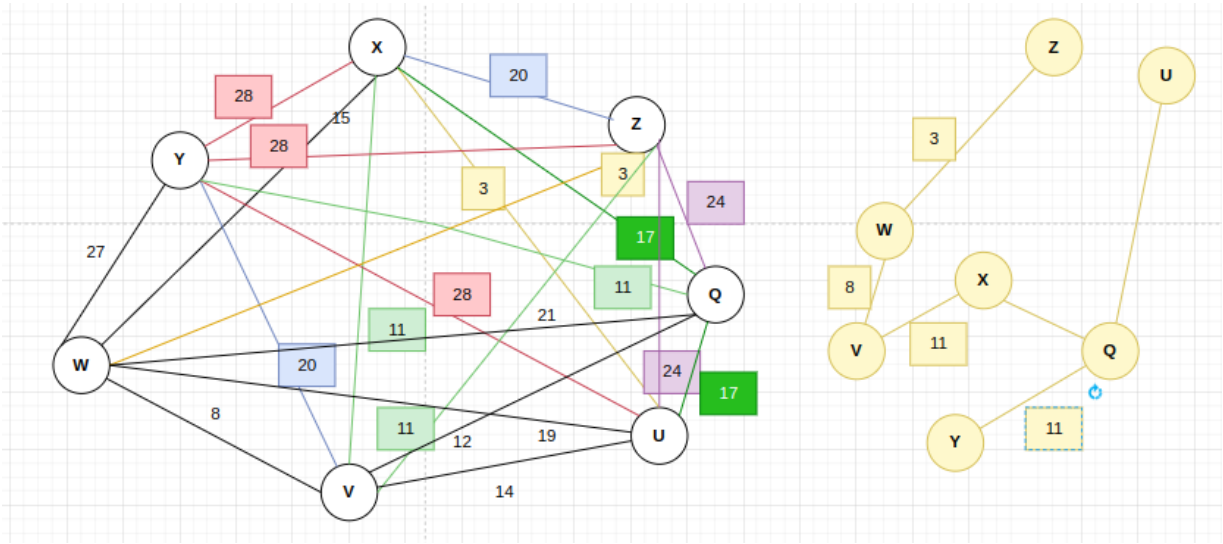
Ahora empezaremos el árbol generador de peso mínimo con el algoritmo de Kruskal.

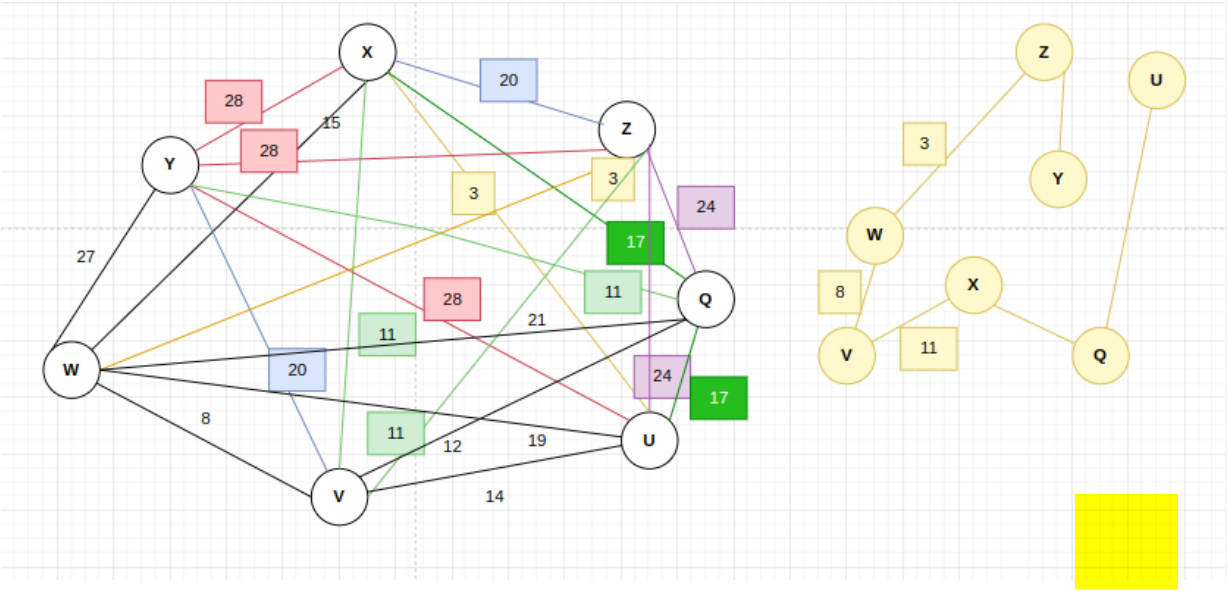




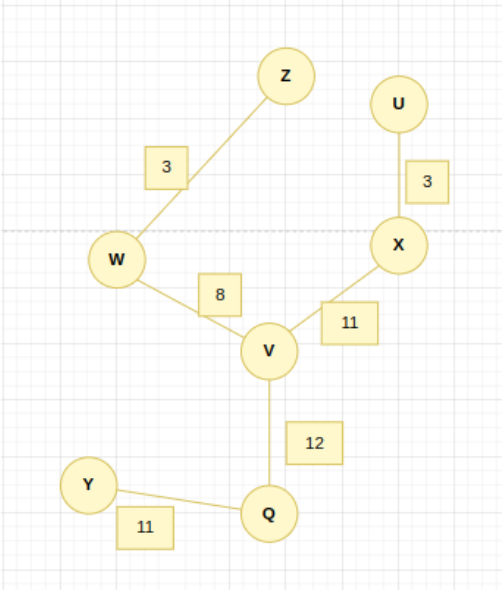

X
Y
Z
W
V
U
Q
20
28
28
15
3
3
24
17
27
28
11
11
21
20
24
17
8
11
12
19
14
Z
U
3
3
W
X
8
11
V
12
Y
Q
11


X
Y
Z
W
V
U
Q
20
28
28
15
3
3
24
17
27
28
11
11
21
20
24
17
8
11
12
19
14
Z
U
3
3
W
X
8
11
V
12
Y
Q
11

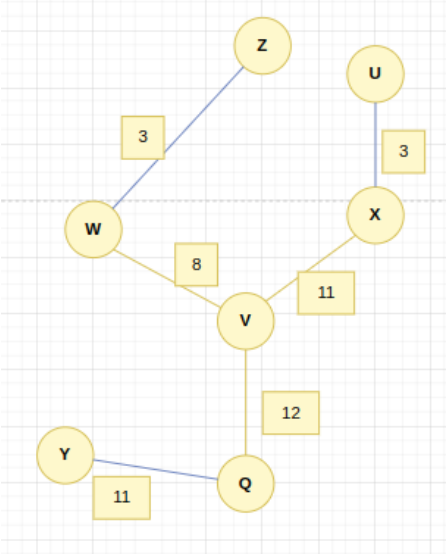

X
Y
Z
W
V
U
Q
20
28
28
15
3
3
24
17
27
28
11
11
21
20
24
17
8
11
12
19
14
Z
U
3
W
X
8
V
11
Q
Y
11

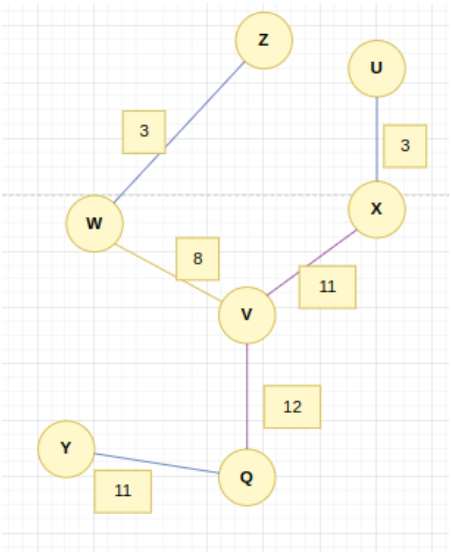


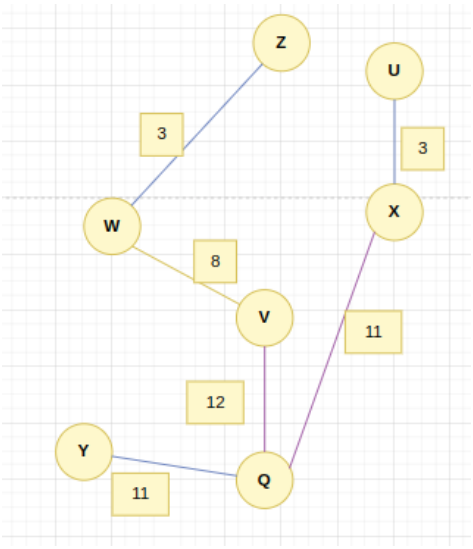
5. **[Opcional]** Aplicar el segundo algoritmo de aproximación dado en las notas y generar tour para el TSP. Deberás detallar cada paso del algoritmo e ilustrar cómo se tomaron los atajos. **Respuesta**
Usaremos la gráfica de peso mínima del ejercicio anterior.

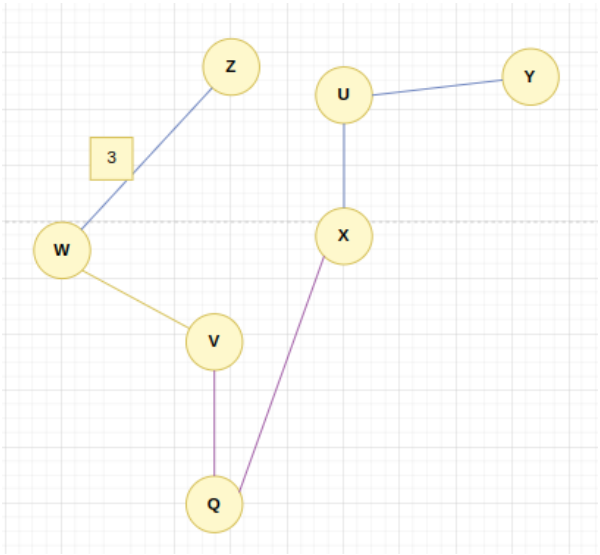


Ahora buscaremos los vértices de grado 1 y les agregaremos a otro vértice.




Z
U
3
3
W
X
8
11
V
12
Y
Q
11


Z
U
3
3
W
X
8
V
11
12
Y
Q
11


Z
U
Y
3
W
X
V
Q