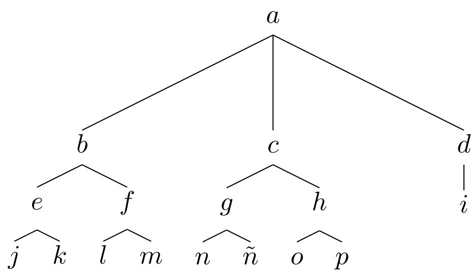
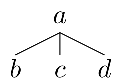
1. El diámetro de una gráfica **G, G=(V,A),** se define como la mayor de las longitudes de las rutas más cortas entre todo par de nodos en la gráfica; es decir:
diam(G)= max d(x,w) : x, w V(G)
donde **d(x,w)** es la longitud de la ruta más corta entre los vértices x y w.
Problema A: Determinar el diámetro de un árbol
a) **Diseñar un algoritmo** de orden $O(|V|)$ que solucione el el **Problema A** y que presente una pareja de vértices cuya ruta más corta sea exactamente el diámetro del árbol. Justificar su respuesta.
b) Presentar una gráfica **G** de al menos 17 vértices y aplicar, con detalle. el algoritmo diseñado en **(a)**
Respuesta
Para este problema utilizaremos **BFS**, ya que se puede almacenar los vértices v y el diámetro d, todo esto solo con la raíz dada. Existe una forma lineal de obtener el máximo de estos valores $\forall v \in V, G$ y que haya sido calculado el nodo máximo que haya sido calculado anterior mente,después con esto ahora si podemos aplicar **BFS** desde el máximo vértice, otorgándonos el diámetro de la gráfica G, es decir, debemos de recorrer la raíz con máximo de v,d y con ayuda de otra variable que cuente el diámetro y aumente cada vez que visitamos un nuevo vértice, además de ir generando tuplas y ir las metiendo en una lista para después mostrarlas como se imprimen.
Proponemos este gráfica:



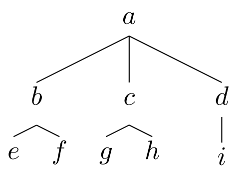
Después aplicamos el BFS y vamos siguiendo el estado actual de cada nodo. Tomando a a como raíz y marcarlo como visitado.

a

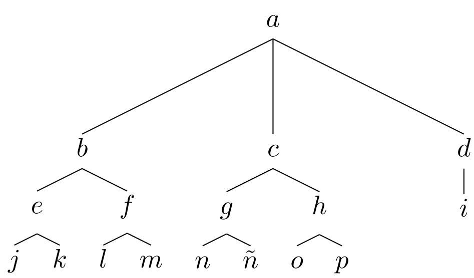
Luego tendríamos a b, c, d visitados y luego tendríamos que aumentar el diámetro de estas a 1.



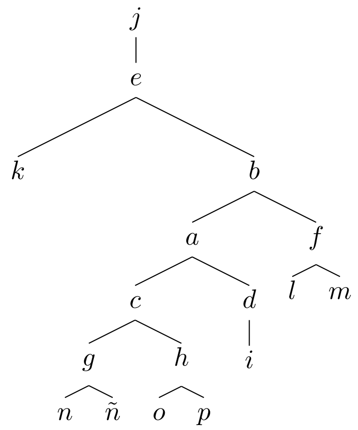
Después los hijos de estos que vendrían siendo e, f, g, h, i serian marcados como visitados y luego aumentaríamos su diámetro a 2



Por ultimo agarramos a los últimos hijos que son $j, k, l, m, ñ, n, o, p$ y estos serían los últimos por visitar, además de aumentar su diámetro a 3.



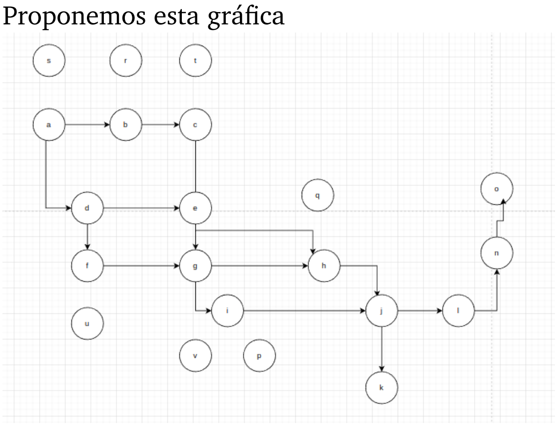
Ahora escogeremos a j con mayor diámetro del diagrama y mostraremos a continuación el nuevo diagrama



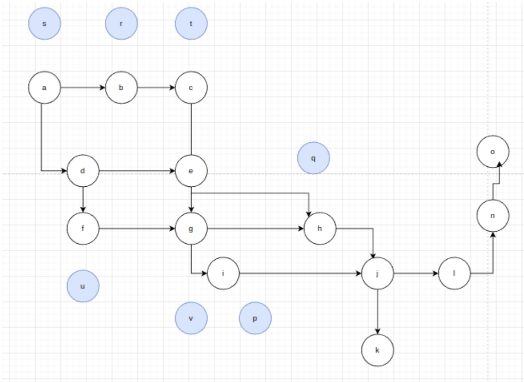
Entonces ahora podemos ver que la gráfica G y se puede ver que el diámetro es de 6, tomando indistintamente cualquier vértices de los últimos elementos $j, k, l, m, n, \tilde{n}, o, p$, propongamos las tuplas de $[(j,e)(e,b)(b,a)(a,c)(c,h)(h,p)]$

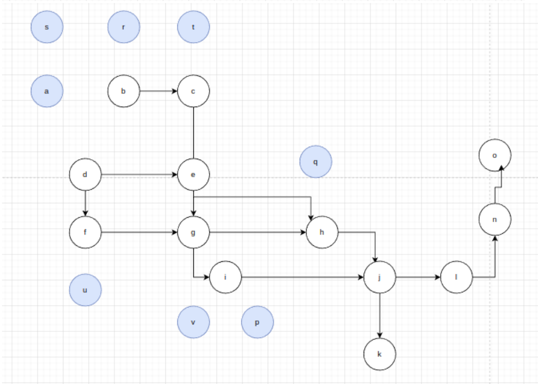
2. Considerar los algoritmos **BFS** y **DFS.** modificar uno de ellos para que acepte como entrada una gráfica no conexa **D.**
Problema B: Determinar el bosque generador de una gráfica disconexa **D.**
a) Diseñar un algoritmo que solucione el **Problema B.**
El algoritmo debe ser capaz de mostrar cada árbol generador (indicando el conjunto de aristas, por ejemplo)
b) Determinar la complejidad del algoritmo propuesto, **indicando** las estructuras de datos utilizadas y su impacto en el desempeño computacional del algoritmo.
c) Presentar una gráfica G disconexa de **al menos:** 17 vértices, 27 aristas y 4 componentes y **aplicar** el algoritmo propuesto a la gráfica presentada. **Respuesta**

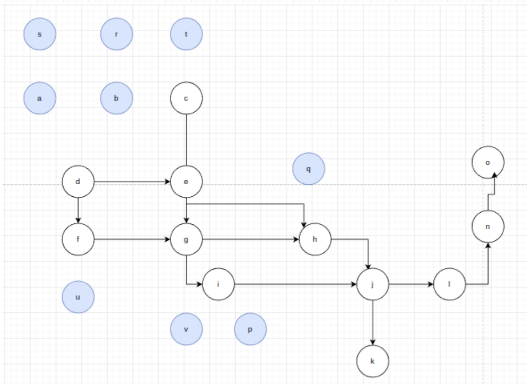
3. **Topological-Sorting (T-S)**
a) Construir una gráfica **G** de al menos 17 y 21 aristas vértices donde **sí** pueda aplicarse Topological-Sorting. Aplique con detalle T-S sobre **G.**
Respuesta
Proponemos esta gráfica

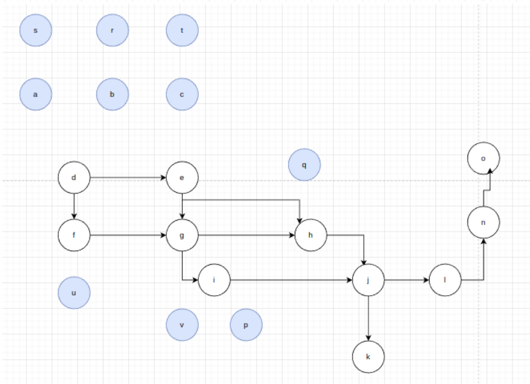


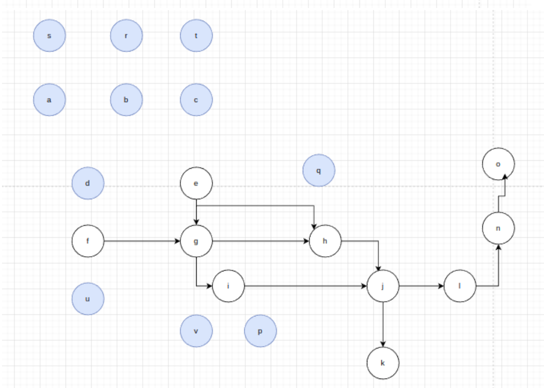
Luego marcamos los vértices visitados de color azul y iremos quitando las aristas también. Como se muestra a continuación

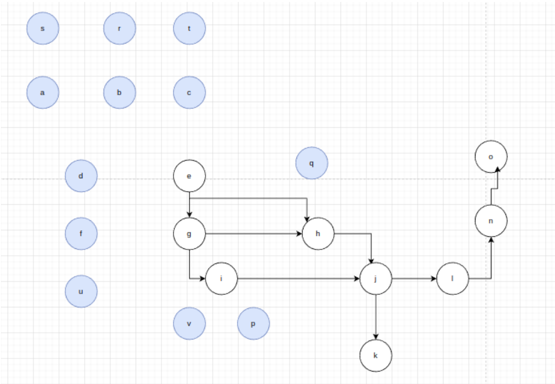

s
r
t
a
b
c
q
o
d
e
h
n
f
g
u
i
j
l
v
p
k

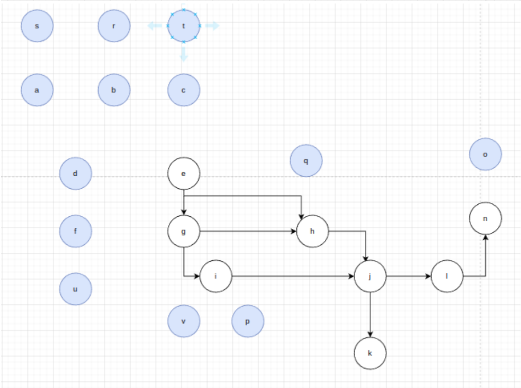

s
r
t
a
b
c
q
o
d
e
h
n
f
g
u
i
j
l
v
p
k

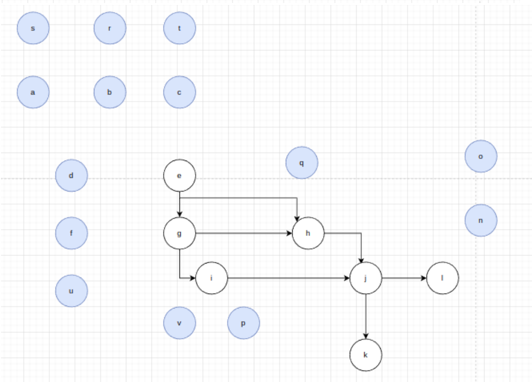

s
r
t
a
b
c
q
o
d
e
h
n
f
g
u
i
j
l
v
p
k

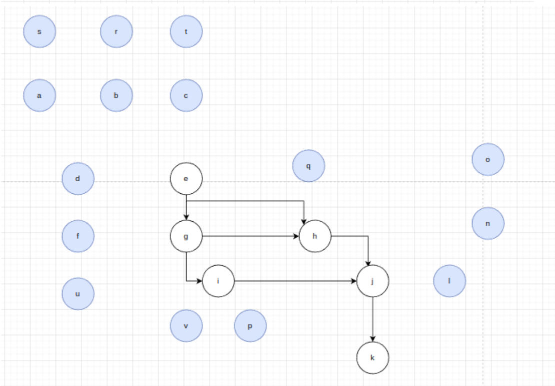

s
r
t
a
b
c
q
o
d
e
h
n
f
g
u
i
j
l
v
p
k

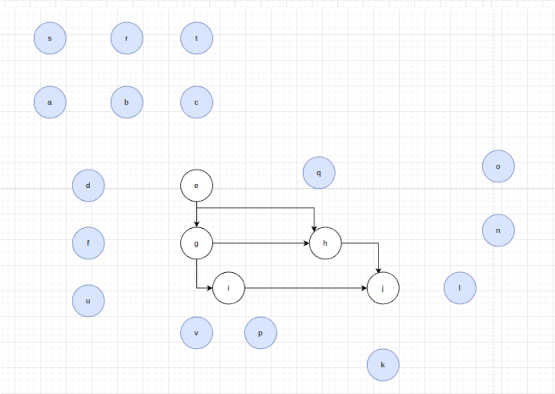

s
r
t
a
b
c
q
o
d
e
h
n
f
g
u
i
j
l
v
p
k

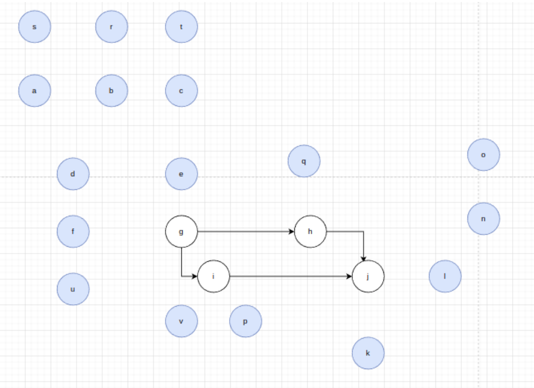
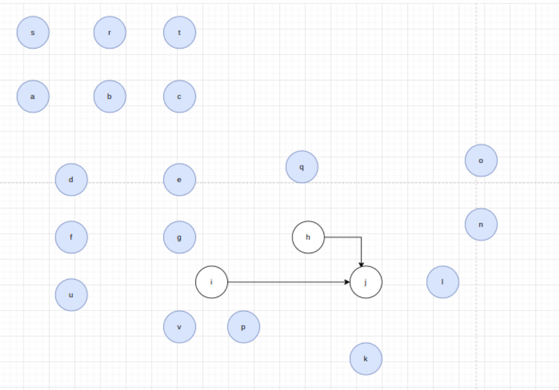
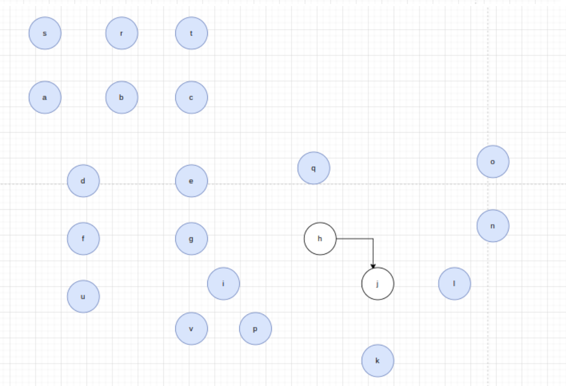
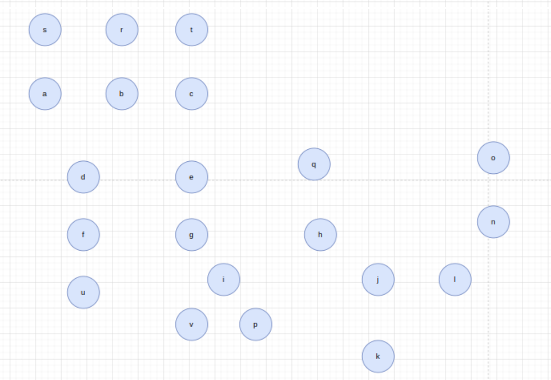

s r t
a b c
d e q o
f g h n
u i j l
v p
k


s r t
a b c
d e q o
f g h n
u i j l
v p
k

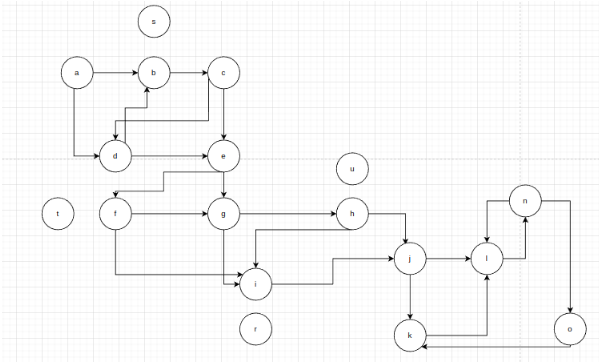

s r t
a b c
d e q o
f g h n
u i j l
v p
k


s r t
a b c
d e q o
f g h n
u i j l
v p
k


s r t
a b c
d e q o
f g h n
u i j l
v p
k

Por ultimo tenemos que su orden topológico es: $[t, r, s, q, u, v, p, a, b, c, d, f, o, n, l, k, e, q, h, j]$
b) Construir una gráfica **G** de al menos 17 vértices y 23 aristas donde **no** pueda aplicarse Topological-Sorting, indique las razones por qué no podría aplicarse T-S
Respuesta



En esta gráfica que se muestra no se puede, debido a que en esta gráfica cuenta con ciclos dirigidos, lo cual hace que no se puede aplicar T-S

4. **[Opcional]** Modifique el algoritmo Topological-Sorting (T-S) de la siguiente manera:
   Suponga que no se sabe si **G=(V,E),** la gráfica dada, tiene o no ciclos dirigidos.
   a) Diseñar un algoritmo que sea capaz de generar la etiquetación de T-S cuando G sea cíclica, además el algoritmo debe ser capaz de identificar el ciclo.
   b) Determinar la complejidad del algoritmo propuesto.
   c) Construir una gráfica G de al menos 17 vértices y 21 arcos que contenga al menos dos ciclos dirigidos y aplique la modificación de T-S presentada en (a).
   Respuesta