

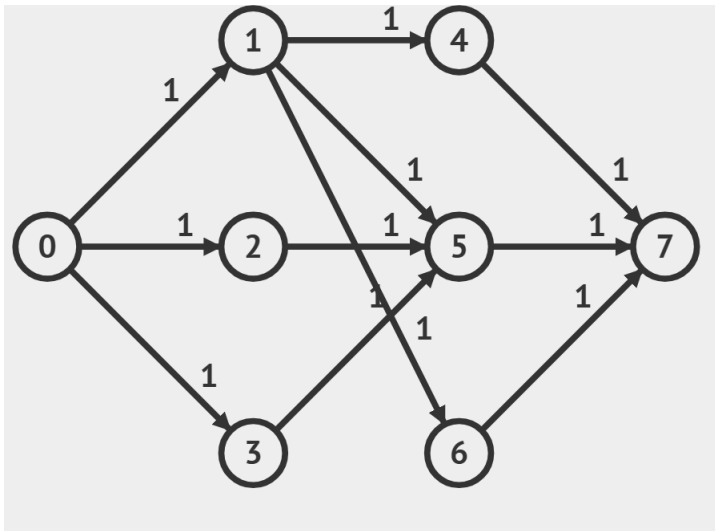
Taller Ford-Fulkerson

1. ¿Cuál es la complejidad temporal de las dos implementaciones?

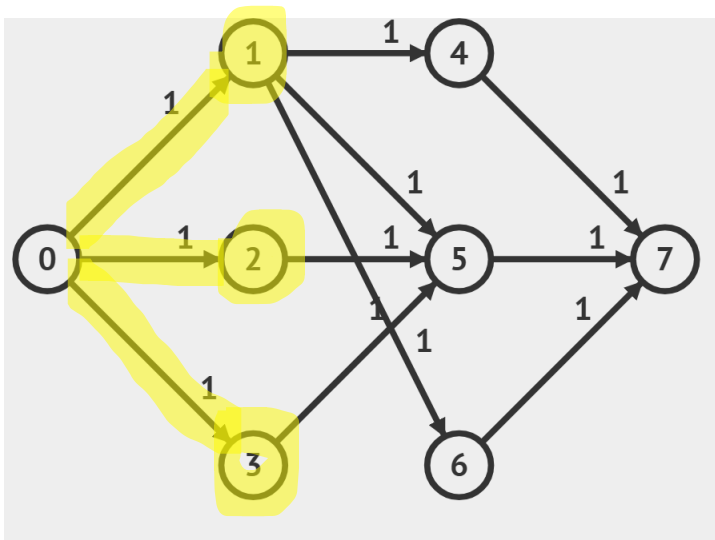
Para ambas implementaciones, en el peor de sus casos la complejidad es de $\mathcal{O}(|F||E|)$ donde $|F|$ es la magnitud del flujo máximo que puede tener la red y $|E|$ son la cantidad de arcos del grafo. Esto es porque en el peor de los casos se agregará al flujo máximo una unidad de material en cada posible camino encontrado.

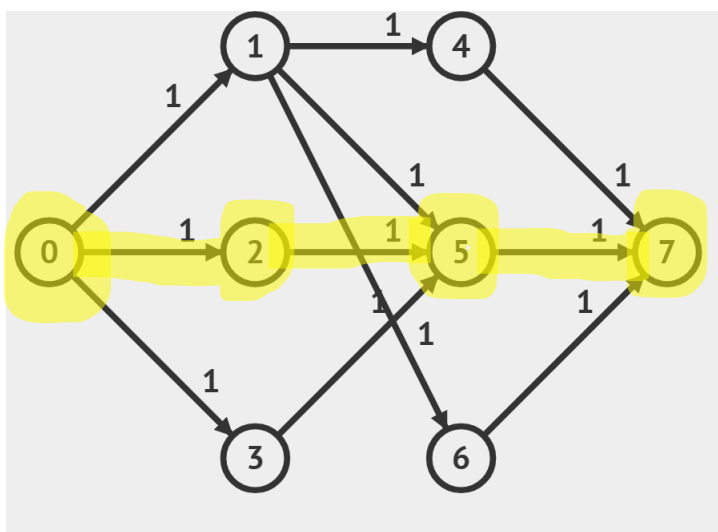
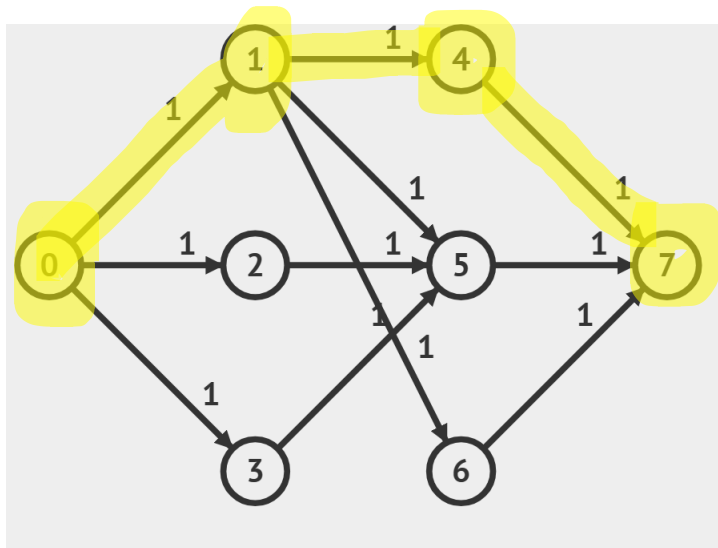
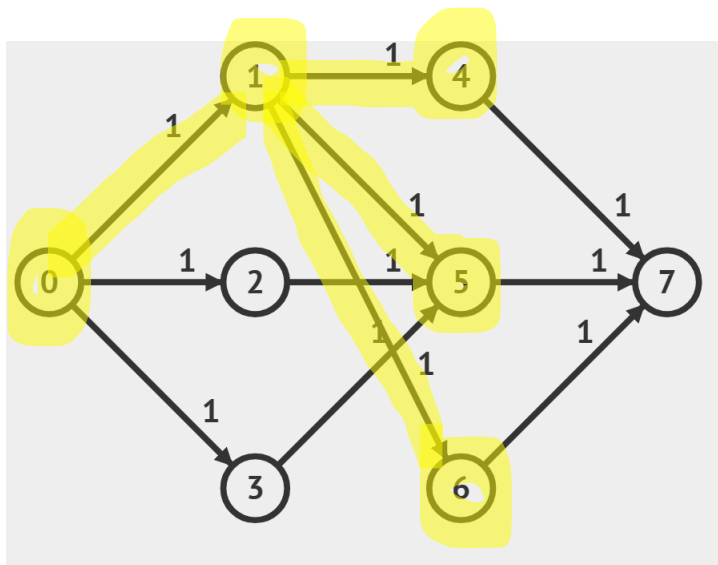
2. Proponga dos redes de flujo (con más de 5 Nodos y más de 5 conexiones) topológicamente distintas donde se evidencie la diferencia en la ejecución de las dos implementaciones.

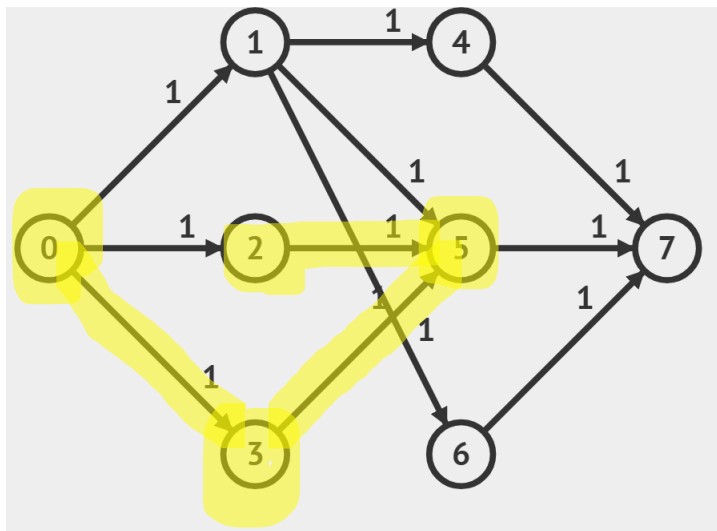
Primera red



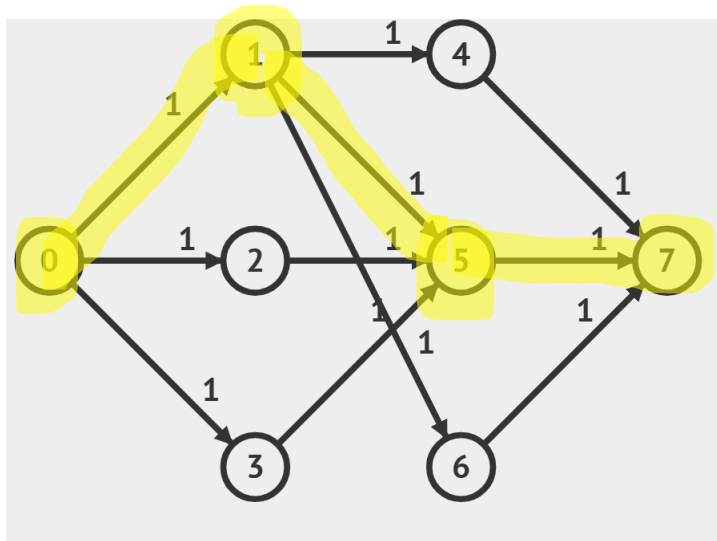
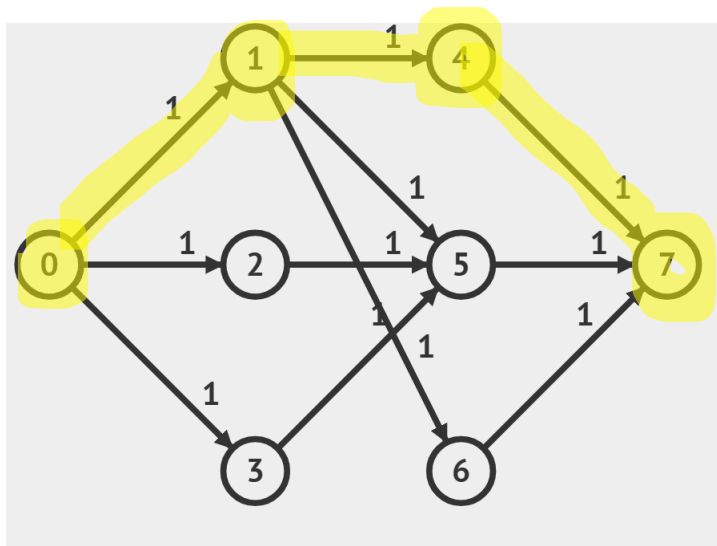
Con BFS:

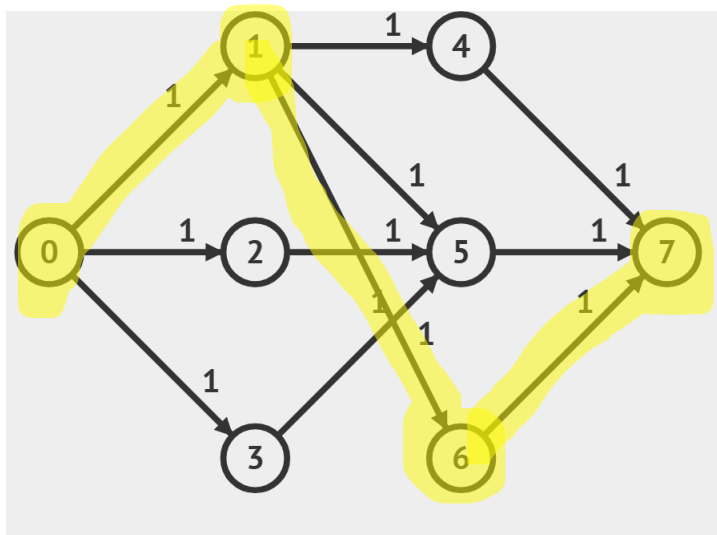




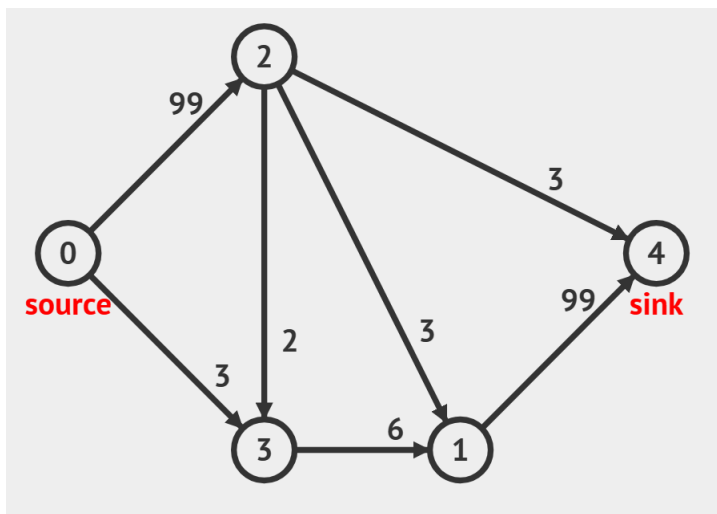


Con DFS

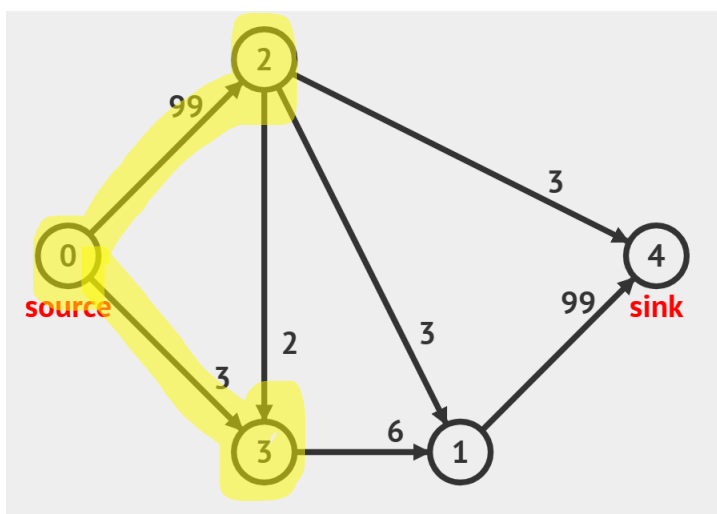


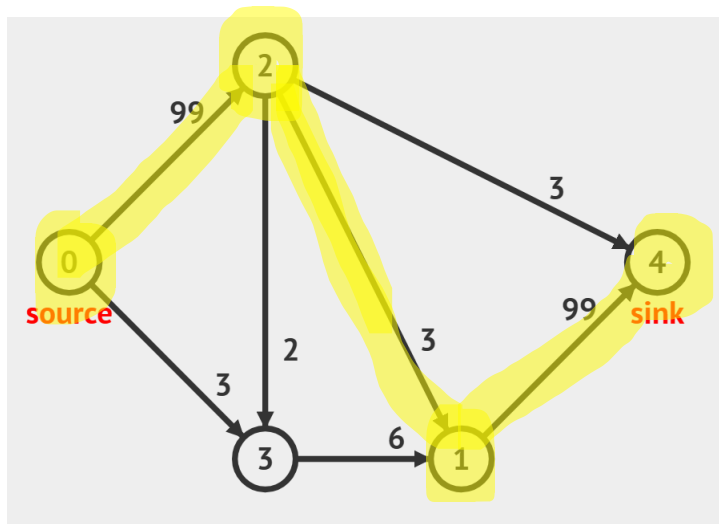
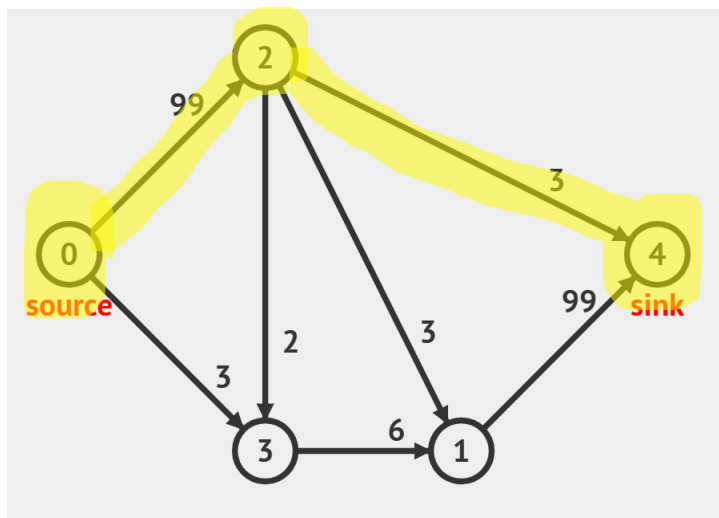
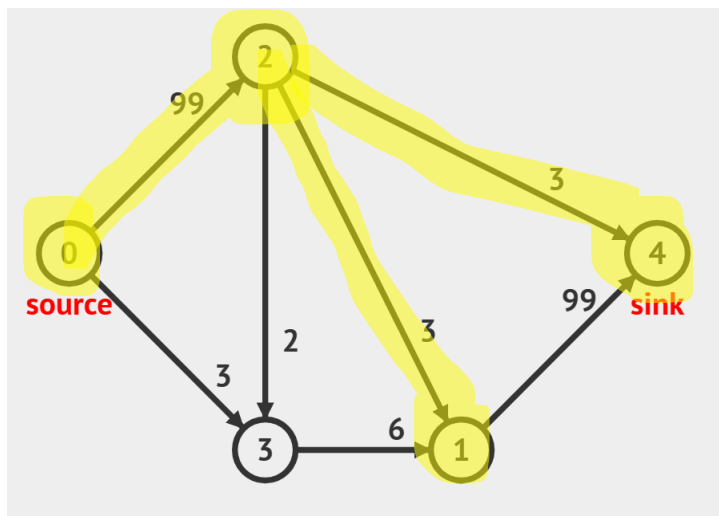


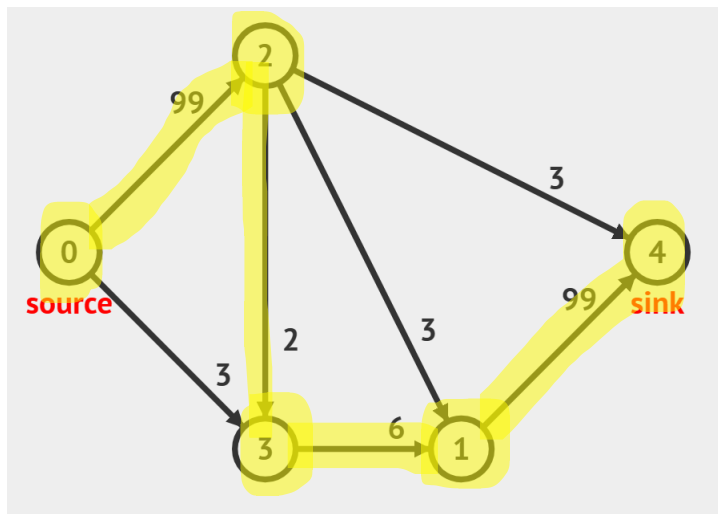
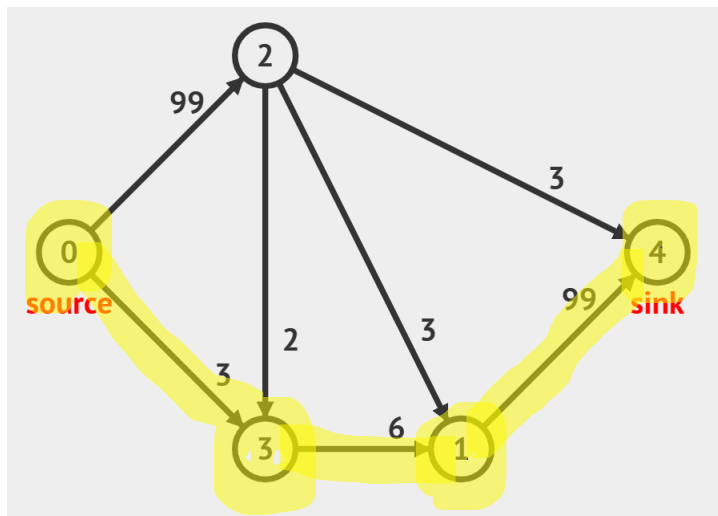
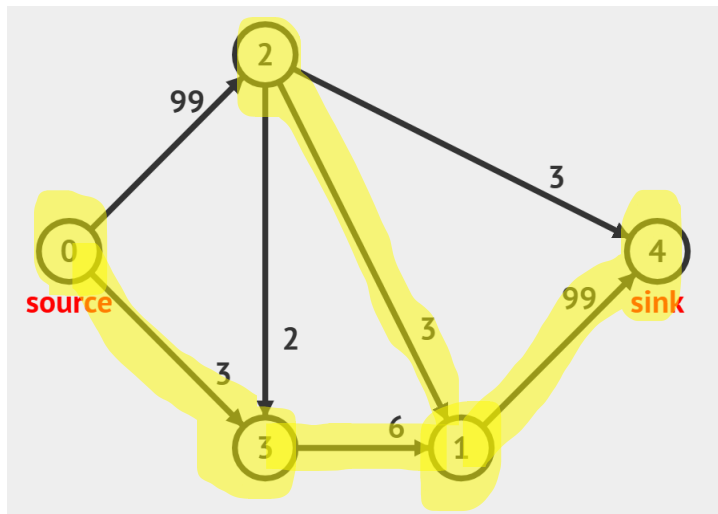
Segunda red



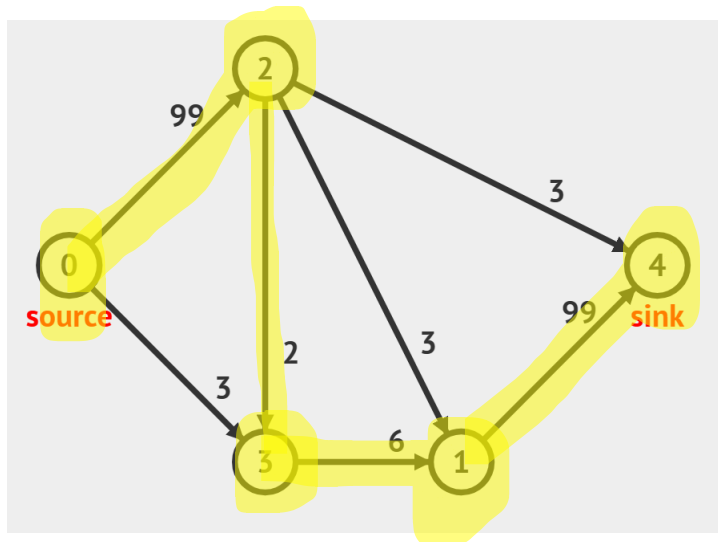
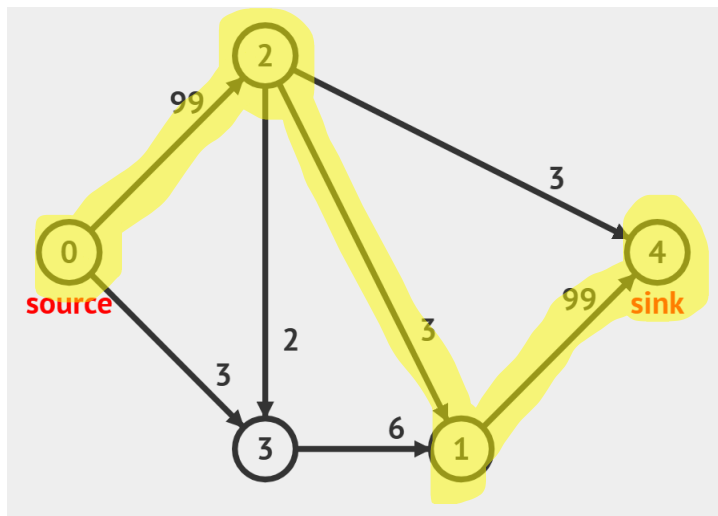
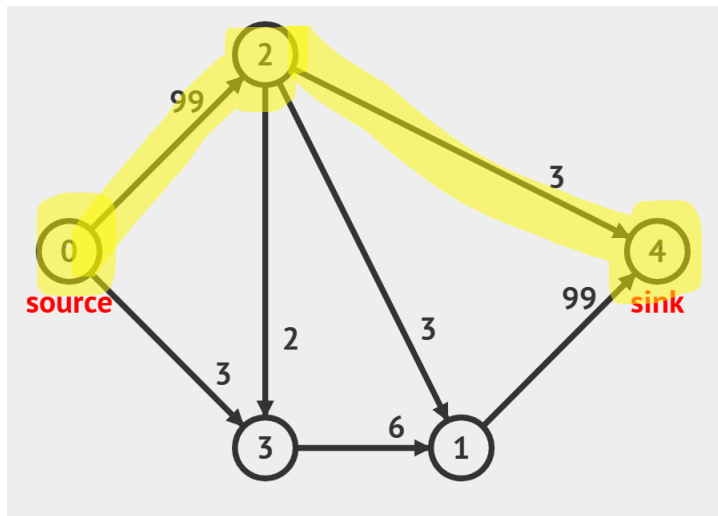
Con BFS:







Con DFS:



Se evidencia la manera en la que BFS explora primero “un nivel” antes de pasar “al otro nivel”, esto puede ser beneficioso o no en algunos casos dependiendo de la estructura específica de la red. La diferencia es

que DFS pasa al otro nivel inmediatamente encuentra un nodo del otro nivel, explicaremos más detalles en la siguiente respuesta.

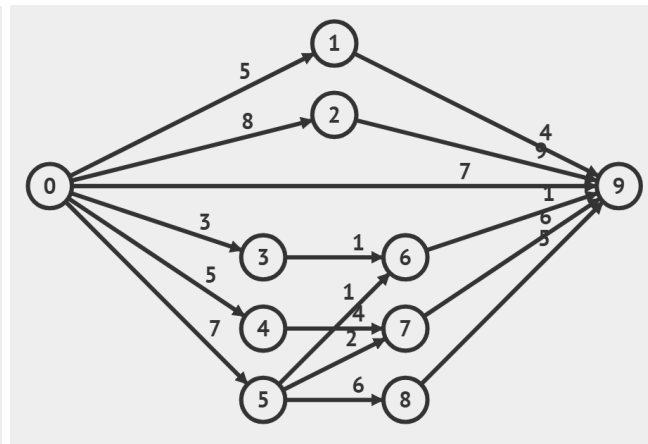
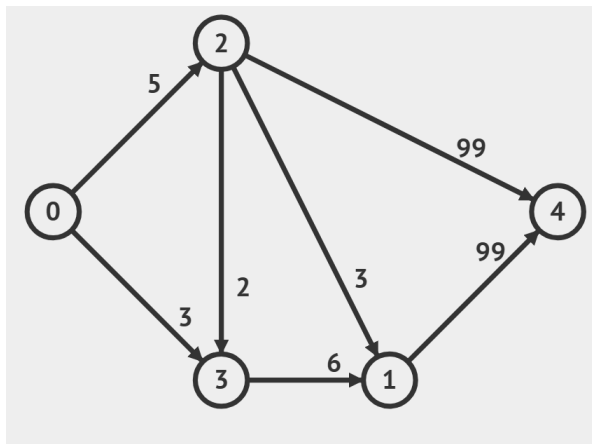
3. Explique cuál de las dos implementaciones es mejor. Si consideran que son idénticas argumente.

La elección de cual implementación es mejor depende mayormente del problema específico, de las características de la red y de los recursos a disposición.

Por ejemplo, si la red es pequeña o tiene una estructura que permite encontrar caminos de aumento cortos con relativa facilidad, BFS tiende a converger más rápido, ya que encuentra caminos más cortos de manera eficiente, pero si la red es grande y compleja, y los caminos de aumento son difíciles de encontrar, DFS podría ser preferible debido a su capacidad para explorar profundamente antes de retroceder. Esto podría conducir a una mejor exploración de la red y, en algunos casos, convergencia más rápida.

Si la definición de mejor fuera la eficiencia espacial BFS por lo general garantiza encontrar el flujo máximo en el menor número de iteraciones, pero puede requerir más espacio en memoria para mantener la cola de nodos a explorar. DFS tiende a ser más eficiente en términos de espacio en memoria, pero puede requerir más iteraciones para converger.

Por lo anterior, nuestra respuesta será que son idénticas en general, pero enfatizando en que una puede ser mejor que otra para casos específicos. A continuación, se muestran dos casos donde el tiempo de ejecución es mejor en uno que en otro, medidos con la librería time de python.



En el caso de la izquierda con BFS dio un tiempo de ejecución de 6.409999332390726e-05 segundos y con DFS de 6.670001312159002e-05 segundos, es decir, para este caso fue mejor BFS.

En el caso de la derecha con BFS dio un tiempo de ejecución de 0.00021349999587982893 segundos y con DFS de 0.00019299998530186713 segundos, es decir, para este caso fue mejor DFS.