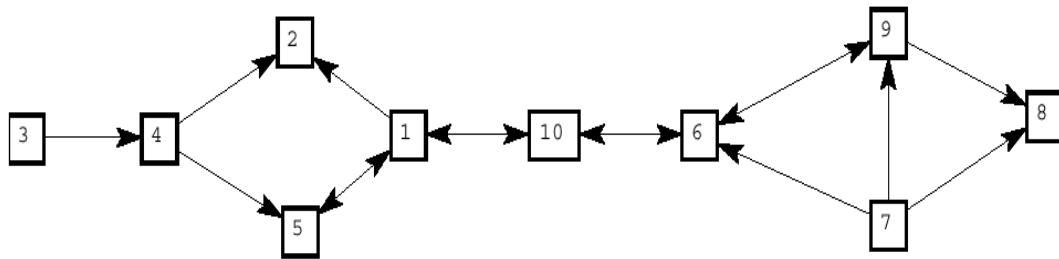


Práctica 1, Análisis de redes con NetworkX (aprox 3h).

En esta práctica vamos a calcular algunos parámetros de un grafo de pequeño tamaño, a continuación, vamos a cargar una red de interacción de proteínas, vamos a calcular sus principales parámetros mediante NetworkX y vamos a compararlos con los de un grafo aleatorio que tenga un número similar de nodos y ramas.

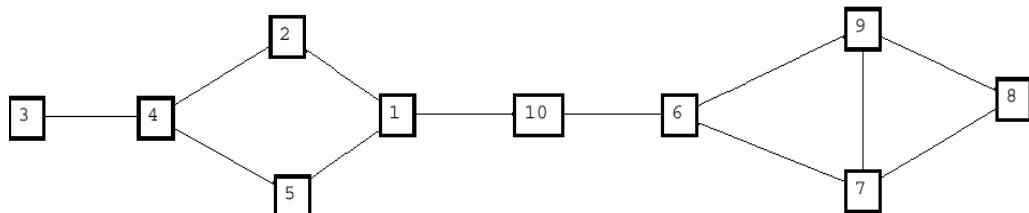
Apartado 1: Este ejercicio debe realizarse con lápiz y papel (o con lápiz y Word) ya que debéis indicar los cálculos que habéis realizado.

1. Representad el siguiente grafo dirigido mediante (a) una matriz de adyacencia y (b) una lista de adyacencia.



2. Responded a las siguientes preguntas:
 - a. ¿Es ponderado?
 - b. ¿Es conexo?
 - c. ¿Es débilmente conexo?
 - d. ¿Cuál es su tamaño y su orden?
 - e. ¿Tiene algún punto de articulación? En caso positivo, indica cual.
 - f. ¿Tiene lazos?
 - g. ¿El grafo tiene algún ciclo? En caso positivo, indica cual
 - h. ¿Existe algún camino entre los nodos 4 y 7? En caso positivo, indica cual es y su longitud.
 - i. ¿Existe algún camino entre los nodos 3 y 9? En caso positivo, indica cual es y su longitud.

Considera ahora el grafo como un grafo no dirigido.



- a. ¿El grafo tiene algún ciclo? En caso positivo, indica cual
- b. ¿Cuál es el mayor valor de k para el cual existe un k-core?
- c. Calcula el índice de clusterización del nodo 10

- d. Calcula el camino característico del nodo 10.
- e. Existe algún cliqué de orden mayor de 2. En caso positivo, indica los nodos que lo componen.

Apartado 2: Análisis de una red de interacción de proteínas mediante NetworkX.

1. Descargad de Moodle el grafo **CaernoElegans-LC_uw.txt**, el grafo contiene una red de interacción de proteínas correspondiente al gusano **Caernobidis Elegans**.
2. El fichero que contiene la red está en formato lista de ramas, por tanto, cargad el grafo en una variable G_CE mediante la función **read_edgelist("CL-LC_uw.txt")**.
3. Obtened e imprimid por la salida el orden y el tamaño del grafo y averiguad si el grafo es dirigido o no. ¿Es un grafo denso o disperso?
4. Cread un grafo aleatorio G_AL que tenga el mismo orden y tamaño que el grafo que acabáis de cargar mediante la función **gnm_random_graph(n,m)**
5. Indica si ambos grafos son conexos.
6. Calcula número de componentes conexas de cada grafo.
7. ¿Cuál es el nodo con mayor grado en cada grafo?
8. ¿Cuál es el nodo con mayor betweenness?
9. ¿Cuál es el nodo con mayor closeness?
10. ¿Cuál es la máxima distancia entre dos nodos del grafo (diámetro del grafo)?

Apartado 3: Distribución del grado de los nodos.

1. Visualizad la distribución del grado de los nodos de ambos grafos.
2. ¿Son iguales las gráficas de distribución de grados de ambos grafos?, ¿Qué conclusión sacas de lo anterior?
3. Dibuja ahora la distribución del grado de los nodos de la red de interacción de proteínas usando escala logarítmica en ambos ejes, añade para ello estas dos líneas de código para cambiar el tipo de escala en cada eje


```
plt.xscale("log", nonposx='clip')
plt.yscale("log", nonposy='clip')
```
4. ¿Qué tipo de gráfica obtienes? ¿Podrías calcular aproximadamente la pendiente de los datos?

Apartado 4: Visualización de los grafos (no esperéis maravillas en este apartado, NetworkX no es muy potente ni muy fácil de usar dibujando).

1. Dibuja ambos grafos ¿observas alguna diferencia entre ellos?
2. Elige algún layout en el cual se pueda observar claramente la diferencia entre ambos grafos.

Exportad ambos grafos a un formato compatible con Gephi, y dibujadlos en Gephi, para ello:

1. Descargad de Moodle el fichero de Java e instaladlo. Exportad ambos grafos en formato GexF mediante **write_gexf(G, path)**.
2. Cargad el fichero con Gephi indicando que se trata de un grafo dirigido.
3. En el panel inferior del lado derecho, donde pone **distribución** (layout), seleccionad alguna distribución (en el caso de la red de proteínas os recomiendo elegir Fruchterman Reingold) y pulsad el botón Ejecutar. Observad lo que ocurre.

Apartado 5: Cálculo de parámetros de los grafos.

1. Calculad, al menos, los siguientes parámetros para ambos grafos
 - a. **degree_centrality(G)** (una vez obtenido el valor para los nodos, debéis hacer el promedio entre todos los nodos, para ello si **d** es el diccionario obtenido que contiene los valores, usad **sum(d.values())** y dividid por el orden del grafo)
 - b. **closeness_centrality(G)** (una vez obtenido el valor para los nodos, debéis hacer el promedio entre todos los nodos)
 - c. **betweenness_centrality(G)** (una vez obtenido el valor para los nodos, debéis hacer el promedio entre todos los nodos)
 - d. **graph_clique_number(G)**
 - e. **average_clustering(G)**
 - f. **El máximo k para el cual existe un k core** (recomendación, usad **core_number(G)**)
 - g. **average_shortest_path_length(G)** de la mayor componente conexa de cada grafo.
2. Construye una tabla comparando los parámetros anteriores para cada uno de los dos grafos ¿Qué diferencias observas entre ambos?

A la vista de los resultados anteriores ¿Qué conclusiones obtienes sobre la red de proteínas analizada

Entregad la práctica (formato pdf o notebook de Jupyter) en la entrega de Moodle.