**A04 Semana 12 - Actividad P01**

**El estudio del mundo de los grafos**

Iniciamos el estudio de la eficiencia y complejidad de los grafos, para ello en esta semana realizaremos lo siguiente:

1. El **“Profesor/Grupo”** dibuja/comparte un ‘**Grafo’** básico (máximo 10 nodos) con el cuál trabajarás la actividad.
2. Construye un archivo de texto como se muestra en la *Fig. 1. “entrada.txt”,* que contenga la descripción del grafo según la siguiente propuesta:

| **100 nodos**  1 A  2 B  3 C  …  9 I  10 J  **50 relaciones**  1 2  2 3  3 4  4 10  …  50 1 |
| --- |

*Fig. 1. “entrada.txt”*

1. Realizado el archivo de **“*entrada.txt*”**, utiliza el operador de direccionamiento en la línea de comando

p.e. → c://construyegrafo.exe **<** entrada.txt **>** salida.txt

1. Codifica un código en *C++* que reciba el archivo y,
   1. Construya una estructura de datos para los nombres de los nodos.
   2. Construya una matriz de adyacencia.
   3. Construya una matriz de incidencias.
   4. Se calcule el tiempo de ejecución.
   5. Al generar el *archivo ‘salida.txt’* deberán estar las dos matrices construidas.
2. Para tener un mejor entendimiento de lo realizado, toma alguna de las matrices y dibuja el grafo auxiliado por algún paquete en *‘Python’*.

Se sugiere utilizar el paquete de Python ‘*NetworkX*’. También es válido que realices una animación de la construcción del grafo en otro lenguaje como ‘*Processing*’ o ‘*PyGame*’.

En tu imagen gráfica muestra los nombres originales de los nodos. Es decir, en lugar de que aparezca un entero que sea una cadena de texto.

1. Solo construye la estructura de las matrices, no se está requiriendo la realización de una clase, ni que se instancien los objetos.
2. Realiza un archivo fuente y ejecutable por cada caso.
3. Apóyate de hacer llamadas a funciones definidas por el programador.
4. La compilación y ejecución se tiene que realizar por la terminal.
5. Utiliza el operador de flujo y la llamada a un archivo de texto con el caso a analizar.
6. Al iniciar la iteración toma el tiempo de inicio, y al finalizar la iteración toma el tiempo final. Utiliza un método de C++.
7. Se sugiere presentar la diferencia en horas, minutos, segundos, milisegundos (00:00:00:00). Revisar la biblioteca ***<ctime>.***
8. En caso de no convertir el tiempo al formato requerido del punto 7 previo, solo respeta utilizar una misma medida de tiempo para todos los análisis posteriores.
9. Observa que no se requiere que hagas un menú o definas objetos.
10. Recuerda siempre tienes que trabajar en una terminal y generar la salida del archivo según tu necesidad.
11. Puedes experimentar haciendo lo mismo en otro lenguaje.