

**“UNIVERSIDAD NACIONAL DE SAN AGUSTÍN”
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERIA DE PRODUCCION Y
SERVICIOS**



DOCENTE: EDITH PAMELA RIVERO TUPAC
ALUMNA: GABRIELA MALENA QUISPE QUISPE
LABORATORIO EDA GRUPO C
TEMA: Grafos

**AREQUIPA-PERÚ
2021**

EJERCICIOS PROPUESTOS:

1. Crear un repositorio en GitHub, donde incluyan la resolución de los ejercicios propuestos y el informe:
2. Implementar el código de Grafo cuya representación sea realizada mediante LISTA DE ADYACENCIA. (3 puntos)

```
Console X
<terminated> Test (13) [Java Application] C:\Users\Gabriela Quispe\.p2\pool\plugins\org.eclipse.
Graph:
piura: camana camana
Iquitos: Ayacucho
aqp: lima
tumbes: chiclayo
Ayacucho: Iquitos
lima: aqp
chiclayo: tumbes
ica: tarapoto
camana: piura piura
tarapoto: ica

The graph has 10 vertex
The graph has 6 edges.
El grafo tiene una arista entre piura y tambien camana.
```

3. Implementar BSF, DFS y Dijkstra con sus respectivos casos de prueba. (5 puntos)

```
Console [X]
<terminated> Recorridos [Java Application] C:\Users\Gabriela Quispe\.p2\pool\plugins\org.eclipse.j
BFS Iterativo:
0
1
3
2
9
5

DFS Iterativo:
0
1
2
9
3
5

DFS Recursivo:
0
1
3
9
2
5
```

Dijkstra:

```
Console [X]
<terminated> Test (14) [Java Application] C:\Users\Gabriela Quispe\.p2\pool\plugins\or
El camino corto es :
S          Node          Distancia
0          0             0
0          1             5
0          2             3
0          3             2
0          4             3
0          5             3
```

4. Solucionar el siguiente ejercicio: (5 puntos)

El grafo de palabras se define de la siguiente manera: cada vértice es una palabra en el idioma Inglés y dos palabras son adyacentes si difieren exactamente en una posición. Por ejemplo, las cords y los corps son adyacentes, mientras que los corps y crops no lo son.

- a) Dibuje el grafo definido por las siguientes palabras: words cords corps coops crops drops drips grips gripe grape graph
b) Mostrar la lista de adyacencia del grafo.

5. Realizar un método en la clase Grafo. Este método permitirá saber si un grafo está incluido en otro. Los parámetros de entrada son 2 grafos y la salida del método es true si hay inclusión y false el caso contrario. (4 puntos)

CUESTIONARIO:

1. ¿Cuántas variantes del algoritmo de Dijkstra hay y cuál es la diferencia entre ellas? (1 puntos)

Algoritmo Clasico Dijkstra	Algoritmo de Dijkstra Filtrado	Algoritmo bidireccional de Dijkstra	Algoritmo de dijkstra filtrado bidireccional
Es ta variante intenta encontrar la ruta más corta (si la hay) entre los nodos de origen y destino dados,	Esta variante intenta encontrar la ruta más corta teniendo en cuenta también una expresión de filtro,	Esta variante del algoritmo de Dijkstra busca la ruta más corta de dos maneras, realiza una búsqueda hacia adelante desde el nodo de origen y hacia atrás desde el nodo de destino.	Esta variante del algoritmo de Dijkstra busca la ruta más corta de dos maneras, realiza una búsqueda hacia adelante desde el nodo de origen y una hacia atrás desde el nodo de destino, al tiempo que agrega las restricciones correspondientes en los bordes dadas por la expresión de filtro.
mientras minimiza la distancia o el costo asociado a cada borde en el gráfico.	que agrega restricciones sobre los bordes potenciales al buscar la ruta más corta entre los nodos de origen y destino.	Si existe la ruta entre los nodos, ambas búsquedas se encontrarán en un punto intermedio.	Si existe la ruta entre los nodos, ambas búsquedas se encontrarán en un punto intermedio.
$O(E + N \log N)$ $O(4 * N)$	$O(E + N \log N)$ $O(4 * N)$	$O(E + N \log N)$ $O(10 * N)$	$O(E + N \log N)$ $O(10 * N)$

2. Investigue sobre los **ALGORITMOS DE CAMINOS MÍNIMOS** e indique, ¿Qué similitudes encuentra, qué diferencias, en qué casos utilizar y porque? (2 puntos)

Algoritmo Bellman-Ford	Algoritmo Floyd-Warshall	Algoritmo de Johnson
este trabaja para encontrar la ruta más corta entre un nodo dado y todos los demás nodos. Aunque es más lento.	Es útil cuando se trata de generar rutas para viajes de varias paradas, ya que calcula la ruta más corta entre todos los nodos relevantes.	Se basa en otros dos algoritmos para determinar la ruta más corta.
Bellman-Ford es capaz de detectar ciclos negativos e informar sobre su existencia.	Calcula la distancia más corta entre cada par de nodos en el gráfico, en lugar de calcular sólo desde un solo nodo.	Funciona mejor con gráficos dispersos, uno con menos aristas, ya que su tiempo de ejecución depende de la cantidad de aristas.
Similar a Dijkstra,	Funciona dividiendo el problema principal en otros más pequeños, luego combina las respuestas para resolver el problema principal del camino más corto.	Cuantos menos bordes, más rápido generará una ruta.