***Problema de Caminos Mínimos – Grupo 1***

***Algoritmo de Dijkstra***

***Problema:*** ***Una empresa de transporte necesita encontrar la ruta más corta entre su depósito y varias tiendas.***

***El objetivo principal es:***

- ***Minimizar el coste de transporte entre puntos (distancia o tiempo).***
- ***Obtener caminos optimizados para planificar rutas eficientemente.***

***Esto se puede modelar mediante un grafo ponderado, donde los nodos representan ubicaciones y las aristas representan rutas con un peso asociado (distancia o tiempo).***

***Representación TDA***

***Para implementar este problema se utilizaron los siguientes TDA:***

| *TDA* | *Descripción* |
|---|---|
| *Nodo<T>* | *Representa un punto (depósito, tienda o pueblo).* |
| *Arista<T>* | *Representa un arco entre dos nodos con un peso asociado.* |
| *GrafoPesado<T>* | *Contiene la lista de nodos y la matriz de adyacencia con los pesos de las aristas.* |
| *Dijkstra* | *Algoritmo que calcula los caminos mínimos desde un nodo origen a todos los demás.* |

***Algoritmo Dijkstra***

***Dijkstra permite calcular los caminos de menor costo desde un nodo origen en toda la red de nodos. El procedimiento básico es:***

1. ***Inicializar todos los nodos con distancia infinita.***
2. ***Seleccionar el nodo origen y asignarle distancia 0.***
3. ***Considerar el nodo seleccionado como actual.***
4. ***Marcar el nodo como visitado y actualizar las distancias de sus vecinos.***
5. ***Repetir el proceso hasta visitar todos los nodos.***

***El resultado final es un arreglo de distancias mínimas desde el nodo origen a cada nodo del grafo.***

***Comparación con otros algoritmos de grafos***

| *Algoritmo* | *Propósito* |
|---|---|
| ***Dijkstra*** | ***Calcula el camino mínimo desde un nodo origen a todos los demás en grafos ponderados sin aristas negativas.*** |
| ***Prim*** | ***Construye un árbol de expansión mínima conectando todos los nodos con el menor costo posible.*** |
| ***Kruskal*** | ***Selecciona los tramos con menor costo hasta unir todos los nodos, evitando ciclos, para formar un árbol mínimo.*** |

***Diferencia clave:*** ***Dijkstra encuentra caminos mínimos desde un origen específico, mientras que Prim y Kruskal construyen árboles de costo mínimo para todo el grafo.***

***Ejemplo de ejecución***

***<u>Se construyó un grafo ponderado con 4 nodos:</u>***

| ***Nodo*** | ***Índice*** |
|---|---|
| ***Depósito*** | ***0*** |
| ***Tienda A*** | ***1*** |
| ***Tienda B*** | ***2*** |
| ***Tienda C*** | ***3*** |

***<u>Aristas con peso:</u>***

| ***Desde*** | ***Hacia*** | ***Peso*** |
|---|---|---|
| ***0*** | ***1*** | ***10*** |
| ***0*** | ***2*** | ***15*** |
| ***1*** | ***3*** | ***12*** |
| ***2*** | ***3*** | ***10*** |

***<u>Resultado del algoritmo Dijkstra desde el Depósito (nodo 0)</u>:***

| ***Nodo*** | ***Distancia mínima desde Depósito*** |
|---|---|
| ***Depósito*** | ***0*** |
| ***Tienda A*** | ***10*** |
| ***Tienda B*** | ***15*** |
| ***Tienda C*** | ***22*** |

***Esto confirma que el algoritmo calcula correctamente los caminos mínimos en un grafo ponderado.***