

INTELIGENCIA ARTIFICIAL

Saul Armando Cuenca Martínez
21110324

**"ÁRBOL DE
MÁXIMO Y
MÍNIMO COSTE
KRUSKAL"**



Parte Teórica

¿Qué es el Árbol de Maximo y Mínimo Coste Kruskal?

El algoritmo de Kruskal es utilizado para encontrar un Árbol de Expansión Mínima en un grafo no dirigido y ponderado. El término "Árbol de Expansión Mínima" se refiere a un subconjunto de aristas del grafo que conecta todos los nodos y tiene el menor peso total posible. Este subconjunto forma un árbol y garantiza que no haya ciclos en el grafo resultante.

¿Para qué sirve el Árbol de Maximo y Mínimo Coste Kruskal?

El algoritmo de Kruskal es útil en una variedad de aplicaciones, como planificación de redes, diseño de circuitos, rutas de transporte, diseño de mapas, gestión de recursos y cualquier situación en la que se necesite conectar nodos o ubicaciones con costos asociados a las conexiones mientras se minimizan los costos totales.

¿Cómo se implementa el Árbol de Maximo y Mínimo Coste Kruskal en el mundo?

Diseño de Circuitos Impresos: En la ingeniería eléctrica y electrónica, el algoritmo de Kruskal se aplica al diseño de circuitos impresos para encontrar la disposición de componentes y conexiones que minimice la longitud de las pistas y, por lo tanto, reduzca el tiempo de viaje de las señales y los costos de producción.

Logística y Transporte: Las empresas de logística y transporte utilizan Kruskal para planificar rutas óptimas para camiones de reparto, servicios de entrega y rutas de transporte público. Esto ayuda a minimizar los costos de transporte y a optimizar la entrega de bienes y servicios.

Optimización de Rutas Aéreas: En la aviación, Kruskal se utiliza para calcular rutas de vuelo óptimas, teniendo en cuenta factores como el consumo de combustible y las condiciones meteorológicas.

Redes de Comunicación de Datos: En redes de comunicación de datos, como las que utilizamos en telecomunicaciones, Kruskal se aplica para establecer conexiones de datos óptimas y garantizar la eficiencia de la transmisión de información.

Planificación de Proyectos de Construcción: En la planificación de proyectos de construcción, Kruskal se utiliza para determinar las conexiones más eficientes entre ubicaciones de construcción, minimizando costos y tiempos de transporte de recursos y materiales.

Gestión de Recursos en Parques Nacionales: En la gestión de parques nacionales y áreas de conservación, Kruskal se usa para planificar rutas de patrulla y caminos de acceso de manera que minimicen el impacto ambiental.

¿Cómo lo implementarías el Árbol de Maximo y Mínimo Coste Kruskal en tu vida?

Redes Sociales y Conexiones Personales: En las redes sociales y al mantener conexiones personales, puedes aplicar conceptos de encontrar conexiones óptimas con amigos, colegas y contactos, lo que puede ayudarte a mejorar tus relaciones y oportunidades.

Planificación Financiera: En la gestión financiera, puedes utilizar la optimización de recursos para asignar tu dinero y tiempo de manera eficiente, garantizando que tus inversiones y gastos estén alineados con tus metas financieras.

Proyectos Personales: Al planificar proyectos personales, como renovaciones en el hogar o la organización de eventos, puedes utilizar principios de Árboles de Expansión Mínima para minimizar costos y recursos utilizados.

Toma de Decisiones Cotidianas: En situaciones donde debas tomar decisiones, puedes considerar la optimización al seleccionar entre opciones disponibles, cómo elegir entre diferentes rutas para llegar a un lugar o tomar decisiones de compra.

Planificación de Actividades Diarias: En tu rutina diaria, puedes aplicar la lógica de priorización y organización basada en la importancia y la eficiencia para aprovechar al máximo tu tiempo y recursos.

Aunque no se aplica directamente un "Árbol de Máximo y Mínimo Costo" en la vida cotidiana, los conceptos relacionados con la optimización y la toma de decisiones eficientes son fundamentales en una variedad de situaciones. Puedes utilizar estas ideas para tomar decisiones informadas y maximizar tus recursos personales, ya sea en la planificación de rutas, en la gestión financiera o en la organización de tareas diarias.

¿Cómo lo implementarías en tu trabajo o tu trabajo de ensueño?

Yo creo que lo mejor es utilizar todas estas herramientas, por que muchas veces las personas quedan estancadas y si tomamos en cuenta como se hace estos te das cuenta que lo puedes implementar en tu vida y en tu trabajo en hacer las cosas mejor posible y ser más productivo, además proponiendo como mejorar la empresa para crecer juntos y ayudando a personas en diferentes sentidos. Lo utilizaría en la tecnología, en transporte, en soluciones de problemas, etc.

Simulador en un programa

```
class DisjointSet:
    def __init__(self, vertices):
        self.parent = [-1] * vertices

    def union(self, root1, root2):
        if root1 != root2:
            self.parent[root1] = root2

    def find(self, vertex):
        if self.parent[vertex] == -1:
            return vertex
        return self.find(self.parent[vertex])

def kruskal(graph):
    num_vertices = len(graph)
    result = []
    edge_index = 0
    sorted_edges = sorted(
        [(graph[i][j], i, j) for i in range(num_vertices) for j in range(i + 1, num_vertices)]
    )
    disjoint_set = DisjointSet(num_vertices)

    while edge_index < num_vertices - 1:
        weight, src, dest = sorted_edges[edge_index]
        edge_index += 1
```

```

edge_index += 1
root1 = disjoint_set.find(src)
root2 = disjoint_set.find(dest)

if root1 != root2:
    result.append((src, dest, weight))
    disjoint_set.union(root1, root2)

return result

# Ejemplo de grafo ponderado en forma de matriz de adyacencia.
graph = [
    [0, 2, 0, 6, 0],
    [2, 0, 3, 8, 5],
    [0, 3, 0, 0, 7],
    [6, 8, 0, 0, 9],
    [0, 5, 7, 9, 0],
]

print("Pasos para construir el arbol de Minimo Coste de Kruskal:")
minimum_spanning_tree = kruskal(graph)
for edge in minimum_spanning_tree:
    src, dest, weight = edge
    print(f"Arista: ({src}-{dest}) - Peso: {weight}")

```

```

Pasos para construir el arbol de Minimo Coste de Kruskal:
Arista: (0-2) - Peso: 0
Arista: (0-4) - Peso: 0
Arista: (2-3) - Peso: 0
Arista: (0-1) - Peso: 2
Press any key to continue . . . |

```

<https://github.com/SaulCuenca/IA-P2/tree/main/P>

R5

References

(n.d.). algoritmo_kruskal [Grafos - software para la construcción, edición y análisis de grafos.]. Retrieved October 26, 2023, from https://arodrigu.webs.upv.es/grafos/doku.php?id=algoritmo_kruskal

Algoritmos de Kruskal y Prim. (2020, March 20).

Wextensible. Retrieved October 26, 2023, from <https://www.wextensible.com/temas/voraces/kruskal-prim.html>