

INSTITUTO TECNOLÓGICO DE CULIACÁN

CARRERA INGENIERIA EN SISTEMAS COMPUTACIONALES



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Materia: Inteligencia Artificial

Hora:18-19

Nombre de Alumno: ABEL ALEJANDRO MARTÍNEZ

LIZÁRRAGA

Nombre del docente: JOSE MARIO RIOS FELIX

Tarea 4- Tabla de Complejidad

Culiacán Sinaloa, 2 de Octubre de 2025.

Tabla comparativa de Algoritmos de búsqueda Inteligente

Criterio	BFS (búsqueda por anchura)	DFS(búsqueda por profundidad)	UCS(búsqueda por Costo uniforme)
Tipo de estrategia	Búsqueda no informada por niveles	Búsqueda no informada en profundidad	Búsqueda no informada basada en costos
Orden de expansión	Expande primero los nodos más cercanos al origen (por nive)	Explora lo más profundo posible antes de retroceder	Expande el nodo con el menor costo acumulado
Garantiza solución optima	Sí, si todos los costos son iguales	No	Sí, siempre que los costos sean positivos
Uso de memoria	Usa mucha memoria ya que almacena muchos nodos en cola	Bajo ya que solo mantiene la ruta actual y nodos por explorar	Alto ya que usa una cola de prioridad y mantiene costos
Completo	Si	No, este puede quedar en ramas infinitas	Si
Eficiencia temporal	Puede ser costoso en problemas grandes	Generalmente rápido, pero puede desviarse mucho	Más lento que BFS y DFS dependiendo de los costos
Aplicaciones típicas	Redes, distancias mínimas, grafos poco profundos	Exploración de laberintos, árboles muy profundos	Problemas donde los costos varían (rutas, mapas, logística)
Estructura usada	Cola	Pila	Cola de prioridad
Ventaja principal	Encuentra la solución más corta en pasos	Bajo consumo de memoria y simple de implementar	Encuentra la ruta de menor costo real
Desventaja principal	Usa mucha memoria	No garantiza la mejor solución	Puede ser lento por el cálculo de costos

Los algoritmos de búsqueda se utilizan en Inteligencia Artificial para encontrar rutas, estados o soluciones dentro de un espacio de posibilidades. Dependiendo del tipo de problema, pueden emplearse estrategias que exploren por niveles, por profundidad o en función del costo acumulado. Su importancia radica en que permiten resolver problemas de planificación, navegación, optimización y toma de decisiones.

- **b – Factor de ramificación:**
Cantidad máxima de nodos hijos que puede generar un nodo dentro del árbol de búsqueda.
- **d – Nivel de la solución:**
Profundidad a la que se encuentra la solución más cercana u óptima.
- **m – Profundidad límite o máxima:**
Mayor nivel que puede alcanzar el árbol. En algunos casos puede no tener un tope definido.
- **C* – Costo mínimo:**
Valor total asociado a la mejor solución posible dentro del espacio de búsqueda.
- **ε – Épsilon:**
Representa el costo positivo más pequeño que puede tener una arista en el grafo.
- **g(n):**
Costo acumulado desde el nodo inicial hasta llegar al nodo n .
- **h(n):**
Estimación heurística del costo restante para llegar desde n hasta el objetivo.

Conclusión

Cada algoritmo de búsqueda tiene fortalezas y limitaciones dependiendo del escenario. BFS es útil cuando se busca una solución corta en términos de pasos; DFS destaca por su bajo uso de memoria; y UCS es ideal cuando se necesita encontrar la ruta más barata en grafos ponderados. La elección del algoritmo depende directamente del tipo de problema y de los recursos disponibles.