



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



**Tecnológico Nacional de México Campus Culiacán**

**Ingeniería en sistemas computacionales**

**Inteligencia Artificial**

**José Enrique Félix Esparragoza**

**Dr. José Mario Ríos Félix**

**6-7 PM**

**Unidad 1**

**Análisis algoritmos de búsqueda**

**24, feb, 2025**

## Análisis algoritmos de búsqueda:

### BFS:

Estado inicial	Total nodos visitados	Total nodos camino	Total tiempo (milisegundos)
{{7, 2, 4}, {5, 0, 6}, {8, 3, 1}}	177809	27	543
{{1, 2, 3}, {4, 0, 5}, {6, 7, 8}}	8011	15	7
{{8, 7, 6}, {5, 4, 3}, {2, 1, 0}}	180899	29	464
{{8, 1, 3}, {4, 0, 2}, {7, 6, 5}}	120426	23	234
{{3, 6, 0}, {1, 8, 4}, {7, 5, 2}}	27096	19	26
{{4, 1, 2}, {3, 0, 5}, {6, 8, 7}}	16974	17	17
{{7, 8, 6}, {2, 4, 3}, {5, 0, 1}}	179434	28	372
{{3, 1, 2}, {5, 4, 6}, {8, 0, 7}}	47942	20	47

Promedio total nodos visitados = 94823.875

Promedio total nodos camino = 22.25

Promedio total tiempo = 213.75

Notas: Tiene un alto costo en memoria y tiempo, especialmente en problemas con grandes espacios de búsqueda. Su ventaja es que encuentra caminos muy cortos.

**DFS:**

<b>Estado inicial</b>	<b>Total nodos visitados</b>	<b>Total nodos camino</b>	<b>Total tiempo (milisegundos)</b>
{{7, 2, 4}, {5, 0, 6}, {8, 3, 1}}	170898	48405	386
{{1, 2, 3}, {4, 0, 5}, {6, 7, 8}}	167811	52285	264
{{8, 7, 6}, {5, 4, 3}, {2, 1, 0}}	181430	2109	366
{{8, 1, 3}, {4, 0, 2}, {7, 6, 5}}	28892	15923	22
{{3, 6, 0}, {1, 8, 4}, {7, 5, 2}}	93815	51181	85
{{4, 1, 2}, {3, 0, 5}, {6, 8, 7}}	86599	47525	106
{{7, 8, 6}, {2, 4, 3}, {5, 0, 1}}	141049	65922	149
{{3, 1, 2}, {5, 4, 6}, {8, 0, 7}}	145343	65692	185

Promedio total nodos visitados = 126,979.625

Promedio total nodos camino = 43,630.25

Promedio total tiempo = 195.375

Notas: Puede ser más rápido en algunos casos, pero su consumo de memoria varía mucho según la estructura del problema y encuentra caminos demasiado largos la mayoría de las veces.

**UCS:**

<b>Estado inicial</b>	<b>Total nodos visitados</b>	<b>Total nodos camino</b>	<b>Total tiempo (milisegundos)</b>
{{7, 2, 4}, {5, 0, 6}, {8, 3, 1}}	55164	551	159
{{1, 2, 3}, {4, 0, 5}, {6, 7, 8}}	276	27	0
{{8, 7, 6}, {5, 4, 3}, {2, 1, 0}}	69556	447	145
{{8, 1, 3}, {4, 0, 2}, {7, 6, 5}}	72835	723	127
{{3, 6, 0}, {1, 8, 4}, {7, 5, 2}}	12894	243	27
{{4, 1, 2}, {3, 0, 5}, {6, 8, 7}}	6068	133	7
{{7, 8, 6}, {2, 4, 3}, {5, 0, 1}}	87611	668	134
{{3, 1, 2}, {5, 4, 6}, {8, 0, 7}}	88599	384	128

Promedio total nodos visitados = 49,125.375

Promedio total nodos camino = 397

Promedio total tiempo = 90.875

Notas: En general es más eficiente que BFS y DFS en tiempo y espacio. Su único pero es que los caminos que encuentra son un poco largos, pero, sin llegar a ser tan largos como los de DFS.

**Búsqueda bidireccional:**

<b>Estado inicial</b>	<b>Total nodos visitados</b>	<b>Total nodos camino</b>	<b>Total tiempo (milisegundos)</b>
{{7, 2, 4}, {5, 0, 6}, {8, 3, 1}}	6103	27	34
{{1, 2, 3}, {4, 0, 5}, {6, 7, 8}}	509	15	2
{{8, 7, 6}, {5, 4, 3}, {2, 1, 0}}	10272	29	29
{{8, 1, 3}, {4, 0, 2}, {7, 6, 5}}	2521	23	5
{{3, 6, 0}, {1, 8, 4}, {7, 5, 2}}	844	19	3
{{4, 1, 2}, {3, 0, 5}, {6, 8, 7}}	783	17	1
{{7, 8, 6}, {2, 4, 3}, {5, 0, 1}}	7326	28	15
{{3, 1, 2}, {5, 4, 6}, {8, 0, 7}}	1153	20	2

Promedio total nodos visitados = 3688.875

Promedio total nodos camino = 22.25

Promedio total tiempo = 11.375

Notas: Es la más eficiente en términos de tiempo y nodos explorados, pero requiere manejar dos búsquedas simultáneas, lo cual lo puede hacer un poco más complicada de implementar. En general, es igual de buena que BFS en encontrar el camino más corto.

**Conclusión:**

El análisis de los algoritmos de búsqueda indica que la búsqueda bidireccional es significativamente más eficiente que las demás opciones. Su capacidad para reducir el número de nodos visitados y el tiempo de ejecución demuestra su superioridad en problemas donde es posible realizar una exploración desde el estado inicial y desde la meta simultáneamente.

En comparación, BFS y DFS pueden explorar una gran cantidad de nodos innecesarios, aumentando el costo computacional. UCS, aunque más eficiente en la búsqueda del camino óptimo, puede ser lento en problemas con muchas conexiones. En contraste, la búsqueda bidireccional logra resolver los mismos problemas con menos recursos, lo que la convierte en una opción altamente efectiva cuando su implementación es viable.