Método	Descripción	Ventajas	Desventajas
binarySearch()	Busca un elemento en una matriz ordenada dividiendo repetidamente la matriz por la mitad y descartando una mitad cada vez.	Es muy eficiente y tiene una complejidad de tiempo de O(log n).	Solo funciona en matrices ordenadas y requiere que la matriz se ordene previamente.
linearSearch()	Busca un elemento en una matriz recorriendo todos los elementos de la matriz uno por uno.	Fácil de implementar y funciona bien en matrices pequeñas.	Ineficiente en matrices grandes, ya que tiene una complejidad de tiempo de O(n).
interpolationSearch()	Es similar a la búsqueda binaria, pero utiliza una fórmula matemática para estimar la posición del elemento buscado en la matriz.	Es más rápido que la búsqueda binaria en matrices que tienen una distribución uniforme de valores.	Puede ser menos eficiente que la búsqueda binaria en matrices con valores repetidos o distribuidos de manera no uniforme.
hashTableSearch()	Almacena los elementos en una tabla hash y busca el elemento utilizando su clave de búsqueda.	Es muy eficiente y tiene una complejidad de tiempo promedio de O(1).	Requiere más memoria que otros métodos y puede ser difícil de implementar correctamente.

treeSearch()	Almacena los elementos en una estructura de árbol y busca el elemento navegando por el árbol.	Es eficiente y tiene una complejidad de tiempo de O(log n).	Requiere más memoria que otros métodos y puede ser difícil de implementar correctamente.
regexSearch()	Busca patrones de texto utilizando expresiones regulares.	Puede ser útil para buscar patrones complejos en texto.	Puede ser menos eficiente que otros métodos para buscar elementos individuales en una matriz.