Datos convencional

Estructura	Peor escenario (50 elementos)			
	Insertar	Busqueda	Acceso	Eliminación
Lista	294600 ns	18700 ns	37900 ns	49000 ns
Pila	9600 ns	29400 ns	43200 ns	4300 ns
Cola	12700 ns	20600 ns	38800 ns	4300 ns

Estructura	Peor escenario (500 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	366700 ns	619500 ns	1628200 ns	2635300 ns
Pila	84400 ns	676300 ns	1590100 ns	35900 ns
Cola	49200 ns	984200 ns	1553500 ns	37100 ns

Estructura	Peor escenario (5000 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	509100 ns	2.31465E7 ns	5.74724E7 ns	6.42157E7 ns
Pila	248600 ns	2.29625E7 ns	5.65756E7 ns	205400 ns
Cola	237500ns	2.51864E7 ns	5.71917E7 ns	210500 ns

Estructura	Peor escenario (50,000 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	2017400 ns	2.3010096E9 ns	5.6273498E9 ns	6.947534E9 ns
Pila	1455400 ns	2.2799025E9 ns	5.7827655E9 ns	1230000 ns
Cola	1482000 ns	2.4170332E9 ns	6.3108987E9 ns	1533400 ns

Datos abstractos

Estructura	Peor escenario (50 elementos)			
	Insertar	Busqueda	Acceso	Eliminación
Lista	430300 ns	79200 ns	31500 ns	33000 ns
Pila	13000 ns	23900 ns	29000 ns	3000 ns
Cola	9400 ns	15700 ns	28400 ns	3100 ns

Estructura	Peor escenario (500 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	322200 ns	1264500 ns	1088800 ns	1131500 ns
Pila	40800 ns	1037200 ns	993900 ns	23000 ns
Cola	45600 ns	583500 ns	1043600 ns	22100 ns

Estructura	Peor escenario (5000 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	945900 ns	2.92458E7 ns	7.04562E7 ns	9.26063E7 ns
Pila	289100 ns	5.06022E7 ns	6.10162E7 ns	168400 ns
Cola	342000 ns	5.10945E7 ns	6.06078E7 ns	179800 ns

Estructura	Peor escenario (50,000 elementos)			
	Inserción	Busqueda	Acceso	Eliminación
Lista	2017400 ns	2.3010096E9 ns	5.6273498E9 ns	6.947534E9 ns
Pila	1455400 ns	2.2799025E9 ns	5.7827655E9 ns	1230000 ns
Cola	1482000 ns	2.4170332E9 ns	6.3108987E9 ns	1533400 ns

Conclusión:

Los tipos de datos abstractos (ADTs, por sus siglas en inglés) proporcionan una abstracción de alto nivel para las operaciones en estructuras de datos, lo que puede llevar a un mejor rendimiento en comparación con operaciones directas en tipos de datos concretos como Integer. Aquí hay algunas razones por las que los ADTs pueden ser más rápidos:

Optimización interna: Los ADTs suelen implementarse utilizando estructuras de datos altamente optimizadas, diseñadas para un rendimiento óptimo en una amplia gama de situaciones. Estas implementaciones pueden aprovechar algoritmos y técnicas de optimización que no están directamente disponibles al usar tipos de datos concretos.

Abstracción de alto nivel: Los ADTs ocultan los detalles de implementación de las estructuras de datos subyacentes, lo que permite a los diseñadores de algoritmos y a los programadores centrarse en el uso y la lógica de la estructura de datos en lugar de preocuparse por los detalles de la implementación. Esta abstracción puede permitir a los compiladores y optimizadores realizar mejoras de rendimiento que no serían posibles con tipos de datos concretos.

Polimorfismo y reutilización de código: Los ADTs se pueden diseñar de manera que sean genéricos y reutilizables en una variedad de contextos. Esto puede conducir a una mayor modularidad y flexibilidad en el diseño del código, lo que a su vez puede permitir optimizaciones de rendimiento en tiempo de compilación y ejecución.

Optimizaciones de compilación: Los compiladores pueden realizar optimizaciones específicas para ADTs genéricos, como la especialización de plantillas en C++ o la inferencia de tipos en Java, que pueden resultar en un código más eficiente en tiempo de ejecución.

En resumen, los tipos de datos abstractos proporcionan una capa de abstracción que puede conducir a un mejor rendimiento en operaciones de estructuras de datos debido a la optimización interna, la abstracción de alto nivel, la reutilización de código, las optimizaciones de compilación y el menor overhead de memoria. Sin embargo, es importante tener en cuenta que el rendimiento real puede variar dependiendo de la implementación específica de los ADTs y de las características del sistema en el que se ejecuten.