

OBSERVACIONES DE LA PRÁCTICA

Juan Pablo Peñaranda Cod 202422285
María Gabriela Herrera Rojas Cod 202510147
Estudiante 3 Cod 202514355

Ambientes de pruebas

	Máquina 1	Máquina 2	Máquina 3
Procesadores	13th Gen Intel(R) Core(TM) i5-13450HX (2.40 GHz)	Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz 2.21 GHz	Intel Core i3-14100
Memoria RAM (GB)	16	16	16
Sistema Operativo	Windows 11 Home Single Language	Windows 11 pro education	Windows 11 home Single Language

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Máquina 1

Resultados para Queue con Array List

Porcentaje de la muestra	enqueue (Array List)	dequeue (Array List)	peek (Array List)
0.50%	0.005	0.002	0.003
5.00%	0.006	0.000	0.002
10.00%	0.004	0.000	0.001
20.00%	0.004	0.002	0.000
30.00%	0.004	0.000	0.002
50.00%	0.007	0.000	0.002
80.00%	0.002	0.000	0.001
100.00%	0.005	0.000	0.001

Resultados para Stack con Array List

Porcentaje de la muestra	push (Array List)	pop (Array List)	top(Linked List)
0.50%	0.001	0.000	0.001
5.00%	0.001	0.000	0.001
10.00%	0.000	0.000	0.000
20.00%	0.000	0.000	0.001
30.00%	0.000	0.000	0.000
50.00%	0.000	0.000	0.001

80.00%	0.000	0.000	0.001
100.00%	0.000	0.000	0.001

Resultados para Queue con Linked List

Porcentaje de la muestra	enqueue (Linked List)	dequeue (Linked List)	peek Linked List)
0.50%	0.003	0.000	0.001
5.00%	0.005	0.000	0.001
10.00%	0.003	0.000	0.001
20.00%	0.003	0.000	0.001
30.00%	0.002	0.000	0.001
50.00%	0.004	0.000	0.001
80.00%	0.006	0.000	0.001
100.00%	0.002	0.000	0.000

Resultados para Stack con Linked List

Porcentaje de la muestra	push (Linked List)	pop (Linked List)	top(Linked List)
0.50%	0.000	0.002	0.001
5.00%	0.001	0.000	0.001
10.00%	0.000	0.000	0.001
20.00%	0.000	0.000	0.001
30.00%	0.000	0.000	0.001
50.00%	0.003	0.000	0.001
80.00%	0.000	0.000	0.001
100.00%	0.000	0.000	0.000

Máquina 2

Resultados para Queue con Array List

Porcentaje de la muestra	enqueue (Array List)	dequeue (Array List)	peek (Array List)
0.50%	0.014	0.006	0.007
5.00%	0.012	0.002	0.006
10.00%	0.012	0.002	0.006
20.00%	0.017	0.004	0.001
30.00%	0.009	0.004	0.001

50.00%	0.012	0.002	0.005
80.00%	0.011	0.002	0.005
100.00%	0.014	0.001	0.006

Resultados para Stack con Array List

Porcentaje de la muestra	push (Array List)	pop (Array List)	top(Array List)
0.50%	0.003	0.002	0.003
5.00%	0.003	0.002	0.002
10.00%	0.002	0.001	0.001
20.00%	0.002	0.001	0.002
30.00%	0.002	0.001	0.002
50.00%	0.004	0.002	0.002
80.00%	0.003	0.002	0.003
100.00%	0.002	0.001	0.002

Resultados para Queue con Linked List

Porcentaje de la muestra	enqueue (Linked List)	dequeue (Linked List)	peek Linked List)
0.50%	0.016	0.003	0.006
5.00%	0.012	0.002	0.005
10.00%	0.015	0.001	0.002
20.00%	0.012	0.002	0.005
30.00%	0.011	0.001	0.003
50.00%	0.010	0.001	0.003
80.00%	0.014	0.001	0.002
100.00%	0.010	0.001	0.002

Resultados para Stack con Linked List

Porcentaje de la muestra	push (Linked List)	pop (Linked List)	top(Linked List)
0.50%	0.005	0.008	0.005
5.00%	0.005	0.002	0.004
10.00%	0.001	0.001	0.002
20.00%	0.003	0.005	0.003
30.00%	0.003	0.002	0.003
50.00%	0.002	0.001	0.003
80.00%	0.001	0.001	0.002

100.00%	0.002	0.001	0.002
---------	-------	-------	-------

Máquina 3

Resultados para Queue con Array List

Resultados para Queue con Array List

Porcentaje de la muestra	enqueue (Array List)	dequeue (Array List)	peek (Array List)
0.50%	0.002	0.000	0.001
5.00%	0.003	0.000	0.001
10.00%	0.003	0.000	0.001
20.00%	0.005	0.001	0.001
30.00%	0.006	0.001	0.001
50.00%	0.008	0.001	0.001
80.00%	0.01	0.001	0.001
100.00%	0.13	0.001	0.001

Resultados para Stack con Array List

Porcentaje de la muestra	push (Array List)	pop (Array List)	top(Linked List)
0.50%	0.002	0.001	0.001
5.00%	0.004	0.003	0.001
10.00%	0.004	0.005	0.003
20.00%	0.005	0.005	0.002
30.00%	0.006	0.002	0.002
50.00%	0.007	0.003	0.003
80.00%	0.009	0.006	0.003
100.00%	0.01	0.008	0.003

Resultados para Queue con Linked List

Porcentaje de la muestra	enqueue (Linked List)	dequeue (Linked List)	peek Linked List)
0.50%	0.003	0.002	0.001
5.00%	0.005	0.003	0.001
10.00%	0.007	0.004	0.001
20.00%	0.01	0.005	0.001
30.00%	0.012	0.006	0.001

50.00%	0.015	0.008	0.001
80.00%	0.012	0.01	0.001
100.00%	0.01	0.012	0.001

Resultados para Stack con Linked List

Porcentaje de la muestra	push (Linked List)	pop (Linked List)	top(Linked List)
0.50%	0.002	0.002	0.001
5.00%	0.004	0.003	0.001
10.00%	0.007	0.004	0.001
20.00%	0.002	0.005	0.001
30.00%	0.002	0.006	0.001
50.00%	0.003	0.007	0.001
80.00%	0.004	0.009	0.001
100.00%	0.005	0.011	0.001

Preguntas de análisis

1. ¿Se observan diferencias significativas entre las implementaciones con ArrayList y LinkedList para las funciones de Queue y Stack? ¿Cuál es más eficiente en cada operación? ¿Por qué una implementación es más rápida en ciertos casos?

R: Si, se observan diferencias entre las implementaciones de array y linked para las funciones, Queue al estar implementado con array, tiene que recorrer toda la información, mientras que, en la linked, se va nodo a nodo verificando la información.

En stack es lo opuesto, se usa la estructura LIFO (Last in First Out) en la que se pueden usar array o linked sin problema, pues será O_1 .

La implementación puede ser más rápida en ciertos casos gracias a la memoria y el procesador, aunque si son elementos FIFO, aunque es importante tener en cuenta que Array se obtiene un acceso más directo a la información y Linked es para insertar o eliminar cosas.

2. ¿Cuándo es preferible usar ArrayList o LinkedList? Si insertamos y eliminamos con frecuencia, ¿qué estructura conviene más? Si accedemos aleatoriamente a elementos, ¿cuál es más eficiente?

R: Es preferible usar ArrayList cuando se necesita acceder rápidamente a los elementos por índice o recorrer la lista de manera secuencial, ya que ofrece acceso aleatorio en tiempo constante. En cambio, es mejor usar LinkedList cuando se realizan inserciones y eliminaciones frecuentes,

especialmente en los extremos de la lista, ya que estas operaciones son muy eficientes, aunque el acceso por índice sea más lento al requerir recorrer la lista nodo por nodo.

3. Durante la ejecución de las pruebas ¿Se presentan anomalías en los tiempos de ejecución que no se explican con la teoría?

R//: En el queue para la máquina 2, los tiempos de queue en array list subían y bajaban casi que por cada muestra. Y al final no llego a obtener un menor tiempo que el de la muestra de 0.5%, lo cual considero una anomalía. Y aunque se supone que para queue es más rápida array list en ciertas máquinas se muestra lo contrario.

4. Complete la siguiente tabla de acuerdo con qué operación es más eficiente en cada implementación (marque con una x la que es más eficiente). Adicionalmente, explique si este comportamiento es acorde con lo enunciado en la teoría. Justifique las respuestas.

		Array List	Linked List	Justificación
QUEUE	Enqueue()	X		
	Dequeue()		X	
	Peek()	X	X	En ambas ver el primer elemento es O1
STACK	Push()	X	X	En ambas estructuras agregar es O1
	Pop()	X	X	En ambas estructuras eliminar un elemento es O1
	Top()	X	X	En ambas estructuras consultar el primer elementos O1