

Listas:

- Secuencia de valores que pueden ser reordenados arbitrariamente en el orden en el que están en la secuencia.
- Suelen ser implementados mediante estructuras enlazadas (listas enlazadas)

Operaciones:

• Agregar elemento al inicio:	<code>appendleft</code>	$O(1)$	• Creación:
• Agregar elemento al final:	<code>append</code>	$O(1)$	<code>from collections import deque</code>
• Eliminar elemento al inicio:	<code>popleft</code>	$O(1)$	<code>l = deque()</code>
• Eliminar elemento al final:	<code>pop</code>	$O(1)$	
• Insertar en posición arbitraria:	<code>insert</code>	$O(K)$	
• Eliminar de posición arbitraria:	<code>erase</code>	$O(K)$	
• Tamaño:	<code>len</code>	$O(1)$	
• Determinar si está vacía:	<code>len(l) > 0</code>	$O(1)$	

Pilas:

- Secuencia de valores en la que solo se tiene acceso por un extremo (topo)

• LIFO: Last In First Out

- Tipicamente se implementan como listas enlazadas

Operaciones:

• push	<code>append</code>	$O(1)$	$O(1)$ amort. $O(1)$ $O(1)$ $O(1)$ $O(1)$
• top	<code>l[-1]</code>	$O(1)$	
• pop	<code>pop</code>	$O(1)$	
• empty	<code>len(l) > 0</code>	$O(1)$	
• size	<code>len</code>	$O(1)$	

Colas:

- Secuencia de valores en la que se tiene acceso por 2 extremos, uno para agregar elementos, otro para remover elementos.

• FIFO: First In First Out

- Tipicamente se implementan como listas enlazadas

Operaciones:

• push	<code>append</code>	$O(1)$
• front	<code>l[0]</code>	$O(1)$
• pop	<code>popleft</code>	$O(1)$
• size	<code>len</code>	$O(1)$
• empty	<code>len(l) > 0</code>	$O(1)$

Tablas de Direcciónamiento Directo:

- * Son estructuras de datos en las que se asocian claves a valores. Estas claves determinan la ubicación en la que se almacenan los valores.
- * Es requerido que en la implementación se pueda obtener de forma eficiente los valores a partir de sus claves.
- * Los arreglos (vectores) y mapas en C++, y las listas nativas y diccionarios de Python son implementaciones de esta estructura de datos.
- * Operaciones:
 - * assign
 - * update
 - * query

Operación	Python (listas nativas)	Python (Diccionarios)
	t = []	t = dict() # t = {}
assign	t.append(10) O(1) amort. t[0] = 10 O(1)	t["hola"] = s O(1) amort.
update	t[0] = 8 O(1)	t["hola"] = s O(1)
query	t[pos] O(1)	t[clave] O(1) clave in t

Colas de Prioridad

- * Corresponden a colas en las que los elementos no salen en el orden en el que entran sino de acuerdo a una prioridad.
- * La prioridad es determinada con respecto a algún criterio que puede ser el valor mismo del elemento o alguna cantidad que se pueda calcular sobre el valor del elemento.
- * Suelen ser implementadas mediante montículos (heaps), árboles binarios de búsqueda平衡二叉树) o otras estructuras que soportan operaciones $O(\log n)$.
- * Operaciones:

* push	heappush(t, 10)	$O(\log n)$	* Creación:
* top	t[0]	$O(1)$	from heapq import heappush, heappop
* pop	heappop(t)	$O(\log n)$	t = []
* size	len(t)	$O(1)$	
* empty	len(t) == 0	$O(1)$	

Conjuntos:

- * Agrupaciones de elementos en las que no hay elementos repetidos y no hay orden para los elementos.
- * Asociación con conjuntos matemáticos
- * Operaciones:

insert	add	O(n)
erase	discard, remove	O(n)
empty	len(s) > 0	O(1)
size	len	O(1)