UNIVERSIDAD NACIONAL
GUILLERMO BROWN

Estructuras de Datos Lineales

Algóritmos y Estructuras de Datos

-00184-

Dr. Diego Agustín Ambrossio
Anl. Sis. Angel Leonardo Bianco

Overview:

Tipos de Datos:

- Definidos por el lenguaje (simples)
- Definidos por el Usuario (complejos)
- Datos Mutables e Inmutables.
- (TAD's) Tipos de Datos Abstractos.

TAD's:

- Tipos de Datos Abstractos:
  - Definidos por el Usuario
  - Tipos de Datos Complejos
  - Modulares
  - Estructuras de Datos Lineales
    - Pilas
    - Colas
    - Listas Enlazadas
    - etc..

Estructuras de Datos Lineales:

Nodos:

Crearemos un nuevo objeto tipo **Nodo**. Este nodo será el elemento de construcción básico de nuestras estructuras de datos lineales.

Ejemplo de la clase Nodo:

```
class Nodo(object):

    def __init__(self, dato=None, prox = None):
        self.dato = dato
        self.prox = prox
    def __str__(self):
        return str(self.dato)
```

Pilas:

El comportamiento de una *Pila* se puede describir mediante la frase “Lo último que se apiló es lo primero que se usa”. Este método se llama **LIFO** (Last In First Out).

Operaciones y Funciones:

- __init__ : Inicializa una pila vacía.
- push (apilar): Agrega un nuevo elemento a la Pila
- pop (desapilar): Remueve el *tope* de la Pila y lo devuelve. Este es el último elemento que se agregó.
- is_empty (está_vacía): Retorna **True** o **False** según si la pila está vacía o no.

Opcionles:

- top : Retorna el *tope* de la Pila (sin removerlo).

Colas:

Todos sabemos lo que es una **Cola** . Este método se llama ****FIFO**** (First In First Out).

Operaciones y Funciones:

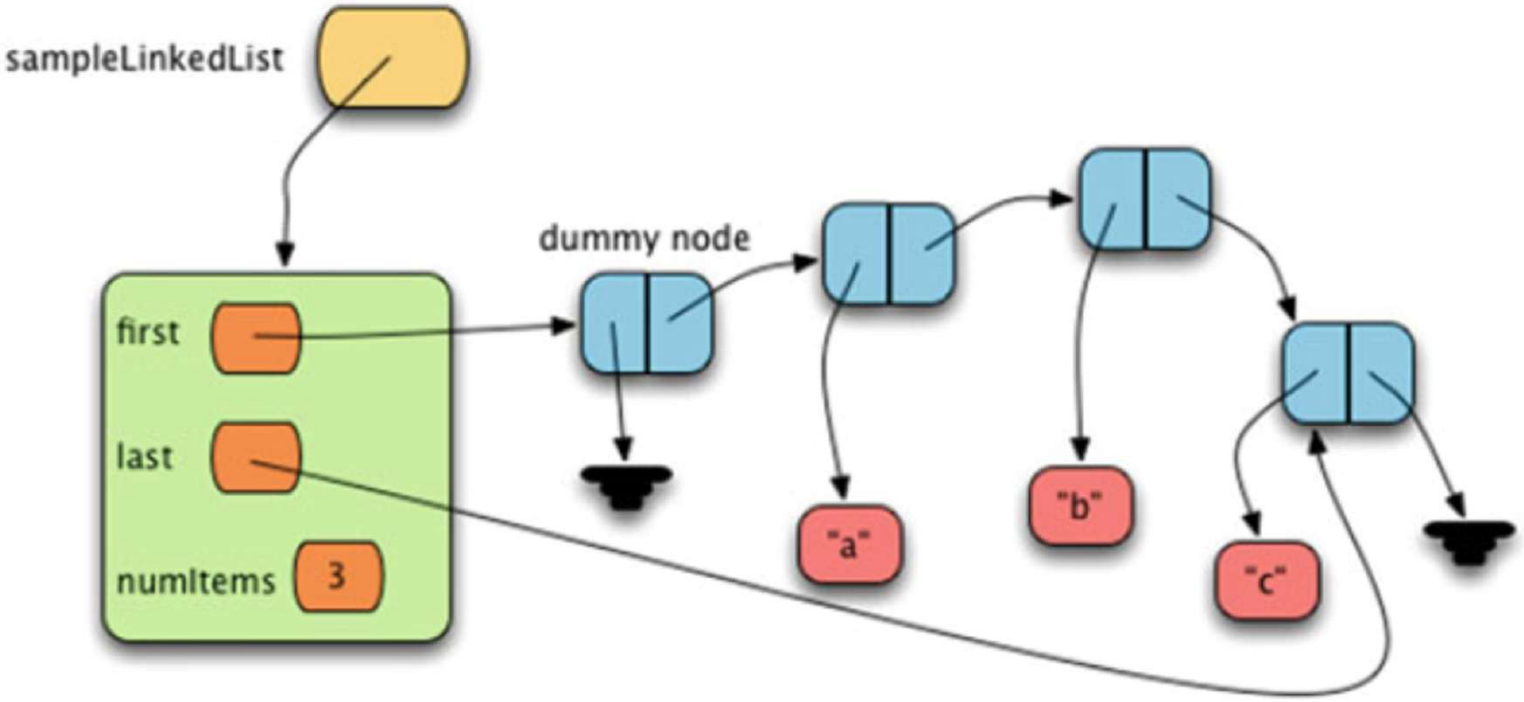
- __init__ : Inicializa una cola vacía.
- enqueue (encolar o push): Agrega un nuevo elemento al final de la Cola
- dequeue (desencolar o pop): Remueve el primer elemento de la Cola y lo devuelve.
- is_empty (está_vacía): Retorna **True** o **False** según si la Cola está vacía o no.

Listas Enlazadas:

- Las listas enlazadas son una secuencia de nodos que se encuentran enlazados cada uno con el siguiente mediante un enlace o puntero.

- Cada elemento (nodo) de una lista enlazada debe tener dos campos: un campo que contiene el valor de ese elemento y un campo que indica la posición del siguiente elemento

Listas Enlazadas:



Listas Enlazadas:

Operaciones y Funciones:

- Inicialización
- Creación
- Recorrido
- Inserción
- Al principio
- Al final
- Borrado
- Búsqueda

Operation	Complexity	Usage	Method
List creation	O(len(y))	x = LinkedList(y)	calls __init__(y)
indexed get	O(n)	a = x[i]	x.__getitem__(i)
indexed set	O(n)	x[i] = a	x.__setitem__(i,a)
concatenate	O(n)	z = x + y	z = x.__add__(y)
append	O(1)	x.append(a)	x.append(a)
insert	O(n)	x.insert(i,e)	x.insert(i,e))
delete	O(n)	del x[i]	x.__delitem__(i)
equality	O(n)	x == y	x.__eq__(y)
iterate	O(n)	for a in x:	x.__iter__()
length	O(1)	len(x)	x.__len__()
membership	O(n)	a in x	x.__contains__(a)
sort	N/A	N/A	N/A

Variantes:

- Listas circulares

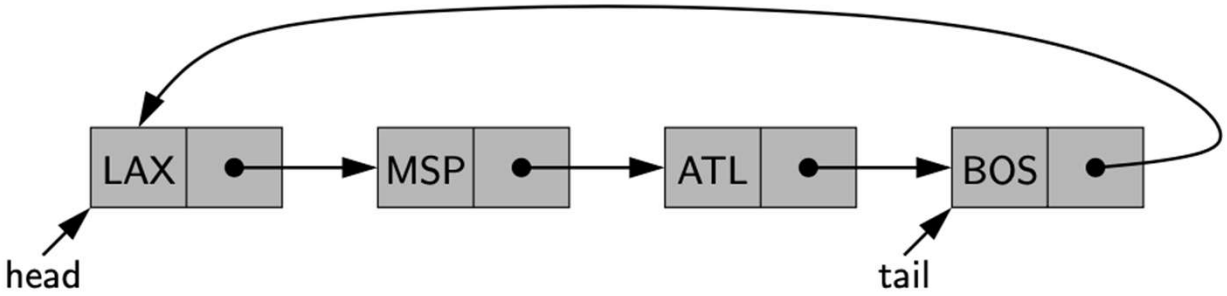


Figure 7.7: Example of a singly linked list with circular structure.

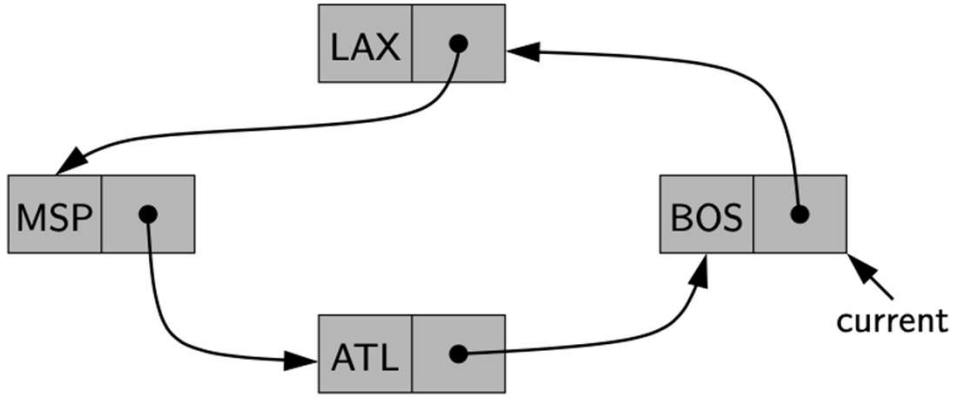


Figure 7.8: Example of a circular linked list, with current denoting a reference to a select node.

Variantes:

- Listas doblemente enlazadas

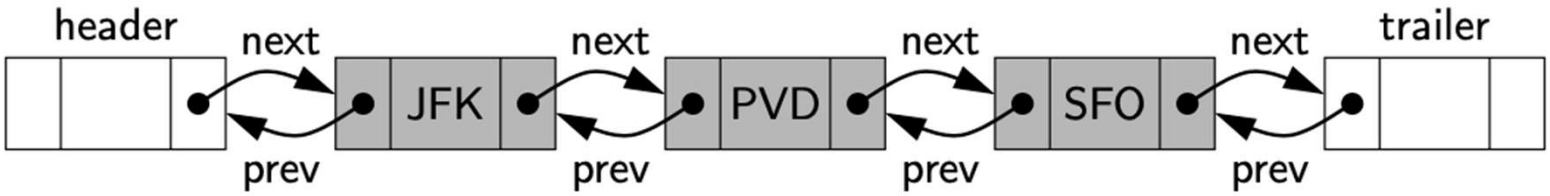


Figure 7.10: A doubly linked list representing the sequence { JFK, PVD, SFO }, using sentinels header and trailer to demarcate the ends of the list.

Otros Tipos de Datos Avanzados:

- Árboles
  - Balanceados
  - de Colores
- Grafos
- HashMaps

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos