1) Linked List Data Structure Dynamic Array Linked List Ly Linear Data Structure (i) Contiguous memory allocation (i) No Contiguous memory allocation * A linear data Structure arranger 10 20 30 40 50 the data in a sequential order, whome elements are Connected - Nentpointer Stores the stores of 1000 other of One after another. Ex: U) Array (ii) Linked List DA-1 (iii) Stack If we want to tir) Queue add new element to the array then Mon Linear Data Structure we double the array * II Organises data in a trienarchical (ii) Lost Node pointer coill always be None or inter Connected way. One element can be connected to (iii) Random access is multiple elements. not possible U) Tree (iv) Insertion Deletion (vi) Searching Operation 1 frequent (ii) Graph Strequent (iii) Heap Linked List Array

(Random Access)

(ir) Trie

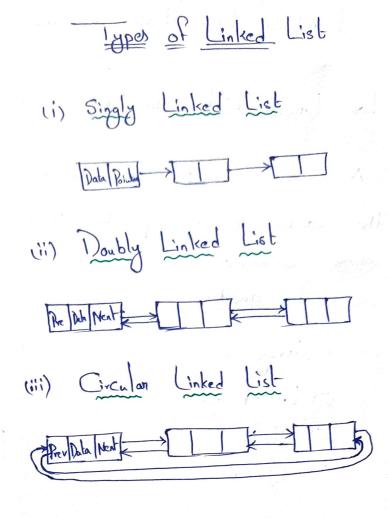
class Linked List:

- O def append (self, value): create a new node add node to end
- 2) def prepend (self, value): Create a new node add node to beginning
- 3 definent (self, value); create a new node and invent node

Jaki a parke for

hory (1)

the the the first of the second of the



and the state of

d halffath it will have

All I do mall significan

The second was the first fitting of the little with the

Clar Node:

def __init__ (self, data):

self. data = data

Self.nextPtr = None

claus Linked List:

def __init__ (self):

Self. head = None

def invent At Begining (self, data):

New_node

new node next pointer should point to the head

Head should point to the new

new-node = Node (data)

new_node.nextPtr = Self. head

selfihead = new-node

def Printlinkedlist (self):

temp - temp - temp

initialize temp to head

Print temp. data

move temp pointer -> temp = temp. nextlb

Repeat this till temp is not None

temp = self. head

while temp:

print (temp. data)

temp = temp. nextPtr

return

def invention At End (self, data):

temp temp (m)

Newshode

Initialize temp to head

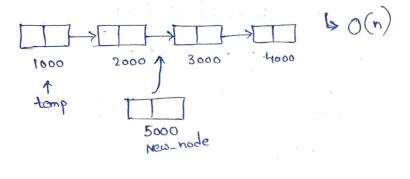
sheck Run loop till temp.nentPb is not

When temp reacher final node, update the temp. nextPh to new node & update temp to new node.

temp: Self. head while temp. nextPtr is not None:
temp = temp. nextPtr

temp. nextPhr = new-node temp = new-node.

def insent After Node (self, Addrew, data):



t A ge

Halla Harada Lings

a hate and the day of the same

they - they

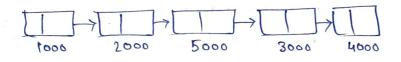
margin make the comment

a part they promised the man had

last their that was not it

o son i fiel a mandar a born and

score water bride has



initialize temp to head

is not equal to given address

If it is equal to given addrew, exit

update new_node nextPh to temp.nextPh.nextPh

And temp. next Ptr. next Ptr to new node

temp= Self. head

cutile temp. next Ptr & Addrew: temp = temp. next Ptr

inchina i villandego a raha

district poor i projek

spiration of the winds

heco-node nentPtr = temp. nextPtr. nextPtr temp. nextPtr. nextPtr = new-node

Linked List - Code Cy with Tail pointer

Class Node:

def __init__ (self, Value):

self. data = value

self . nextPtr = None

class Linked List:

def -- init -- (self, value):

self new_node = Node (value)

self. head = new-node

Self. tail = new_node self. length = 0 Head tail

data Points

(i) Append a node

def append (self, value):

Solf new_node = Value Node (Value)

elle: Self. tail nentitr = new_node

Self . tail = new-node self . length += 1

- if self-head is None:

self. head = new-node

Self. tail = new_node xt.legt=1

(11) Pop a node from end

Edge Caren

(i) Empty Linked List

(ii) Only Single Element

def pop (self):

7 self. head is None

if self. length = = 0:

No Elementa

None

temp = self. head pre = self. head while (temp. nent);

self. Pre = temp

temp = temp. nent_Ptr

Self. tail = pre

self. tail nent. Pt = None

self. length -= 1

if self. length = = 0: After popping self. head = None | last element

Self. tail = None

return temp

(iii) Prepending

def prepend (self, value):

Self. new-node: Node (value)

if self. length = = 0: self. head = new-node self. tail = new-node

che: new-node. nent = self. head self. head = new-node

self. length += 1 return True

(in) Pop Fint

- Ege Care

 (i) Only one node

 (ii) No noder

der pop-first (self):

if self. length = = 0: return None

temp = Self. head

Self. head = Self. head. nent-ptr

temp. hent-pt = None self. length -= 1

if self. length == 0: self. lail = None

return temp

: Konney (9192) - Strat - Fish

(some) small is shown with the

propried a pastro da

okar war i tid i lada O'a digada Sada

spar o torong A vii

: (outer [11.4) brough The

(to be short suring - singe case have

Sport in and + + + 1 throng . Since of the state

show went a link this

complete booking the

swam wom based that

to the second come of had allow