

int main ()

head = 1000

{ struct node * head
~~struct node * head~~ = malloc (size of (struct node));

head -> data = 10;

head ptr = NULL.

struct node * tail = malloc (size (struct node));

tail -> data = 20;

tail -> ptr = NULL

head -> ptr = tail

* find TP for $F(n) = \begin{cases} T(n/2) + 1 & n > 1 \\ 1 & n = 1 \end{cases}$

$$T(n) = T(n/2) + 1$$

$$= T\left(\frac{n/2}{2}\right) + 1 + 1$$

$$= T(n/4) + 2$$

$$= T(n/4/2) + 2 + 1$$

$$= T(n/8) + 3$$

$$\vdots$$
$$= T(n/2^3) + 3$$

$$= T(n/2^k) + k$$

$$n/2^k = 1$$

$$n = 2^k$$

$$k = (\log_2 n)$$

$$T(n) = \begin{cases} T(n/2) + n^2 & n > 1 \\ n^2 & n = 0 \end{cases}$$

* Reverse given list of elements [1, 2, 3, 4, 5] using single linked list.

* Count no. of nodes in the linked list. Delete

* Delete entire linked list. [1, 2, 3, 4, 5]

*

18/1/23

$$T(n) = aT(n/b) + O(n^k \log^p n)$$

1)

$$i) a > b^k = O(n^{\log_b a})$$

$$T(n) = 2T(n-1) + 1$$

$$ii) a = b^k$$

$$T(n) = \begin{cases} 2T(n/2) + n & n > 1 \\ 1 & n = 1 \end{cases}$$

2)

$$i) P > -1 \Rightarrow O(n^{\log_b a} \log^{P+1} n)$$

$$ii) P = -1 \Rightarrow O(n^{\log_b a} \log \log(n))$$

$$iii) P < -1 \Rightarrow O(n^{\log_b a})$$

$$3) a < b^k$$

$$i) P > 0 \Rightarrow O(n^k \log^P n)$$

$$ii) P < 0 \Rightarrow O(n^k)$$

Program to create structure for doubly linked list.

Program name

```

{
    struct node * rptr;

    int data;

    struct node * lptr;
}
    
```

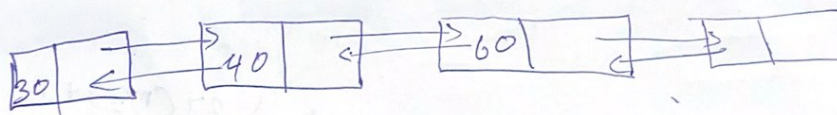
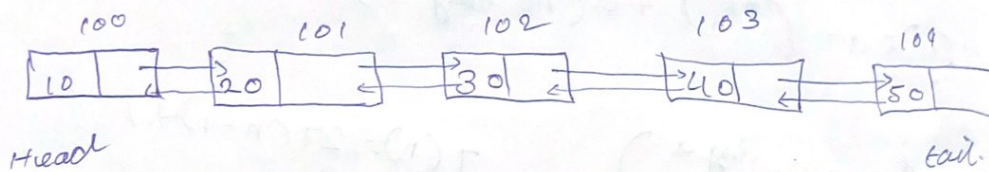
HEAD

| | |
|------|------|
| DATA | LINK |
|------|------|

head → data = NULL

head → link = tail

tail → link = NULL



struct Node * head = malloc (size (struct node))

head → data = 10;

head → ^{left} = NULL

head → Right = NULL

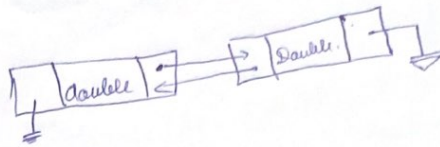
tail → data = Right

tail → Left → NULL

tail → Right → NULL

Head \rightarrow Right = Tail

Tail \rightarrow Left = Head.



$$2 + 8 \times 4 \Rightarrow 14$$

$$2 + (8 \times 4) \Rightarrow 14$$

$$(2 + 8) \times 4 \Rightarrow 20$$

What ADT IS USED TO SOLVE
STACK ADT.