

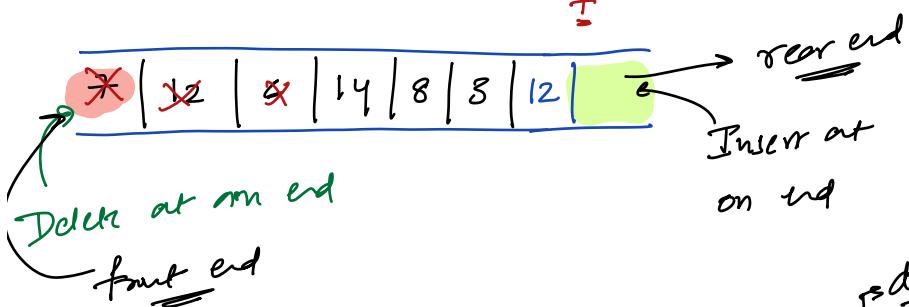
Today's Content:

- Queue Implementation using linked list ✓
- Implement Queue using Stacks
- Reverse Queue
- Reverse first k elements in Queue
- Generate N^{th} Number using 1, 2, 3
- N^{th} Perfect Number
 - ↳ even length
 - Palindrome
 - digit (1, 2)
- DeQueue Basics
- { Plan in Grey window of size = k [if Time permits] }

Queues: FIFO \Rightarrow first in first out Real: ATM / Toll booth
general

Computer: Printer
CPU scheduling: { Queue }
Play songs

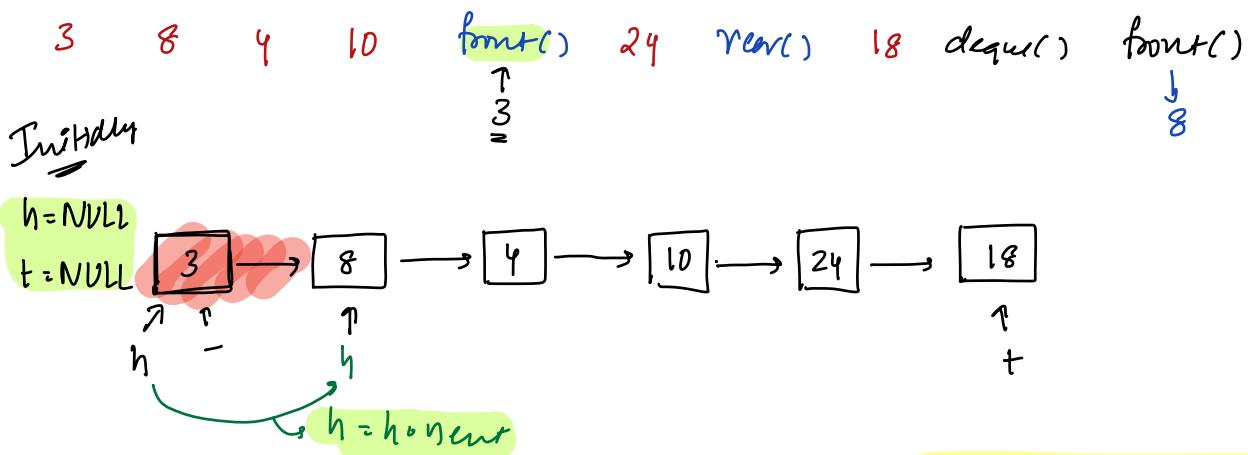
Data: 7 12 6 14 ↑ 8 3 ↑ 12 ↑



Functions in Queue: Enqueue(): 7 8 9 ↑ T front() 10 19 rear()

- | | |
|--|--|
| <ol style="list-style-type: none"> <u>Enqueue(n)</u> <u>Dequeue()</u> <u>rear()</u>: { Return element we have just inserted } <u>front()</u>: { Return element we are going to delete }
 { oldest inserted element } <u>size()</u>: { Number of elements } | ↑
<u>7</u> <u>8</u> <u>9</u> <u>10</u> <u>19</u>
↓ |
|--|--|

Implementation → Using SLL



enqueue(n)

```

Node n = new Node(n)
if(h == NULL) { h = n, t = n }
else {
    t.next = n
    t = n
}

```

front()

```

if(h == NULL) { return Error }

```

return (h.data)

Queue library

TC: O(1)

size(): Using size variable

dequeue() {

```

if(h == NULL) { return Error }
Node tc = h; h = h.next
tc.next = NULL; fruct(t)

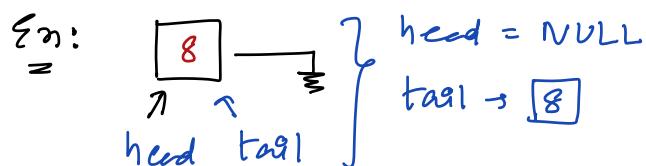
```

rear()

```

if(h == NULL) { return Error }
return t.data

```



enqueue(10):

\uparrow
10

\uparrow
head tail

//

Queue <int> q
↓
We can store integer

→ {
q.enqueue(n); }
q.dequeue(); }
q.front(); q.rear(); q.size(); }
}

Tc: O(1)

2Q) Given a Queue, Revert the data in Queue
(Extra Space)

Sol:

q : 8 10 9 12 6 4 2

Pdear1: not working

delete q pscrr

: order not changing

Pdear2:

{ delete q pscrr in ar[]

reverse ar[]

pinsert in que

Pdear3: Insert element in st

queue

while (q->size() > 0)

{ push (front element of que)
Delete from que

2
4
6
12
9
10
8

2 4 6 12 9 10 8

while (st.size() > 0)

{ pinsert (top element of stack)
pop element from stack

2
4
6
12
9
10
8

TC: $\Theta(N)$

3Q) Given a Queue, Reverse only first k Elements in it.

Note: Do with 1 Stack { Lot of interviews }

Ex: $q:$ 8 8 x 10 14 9 11 5 1 6

$k=4$

Idea:

Step 1: (Delete k Elements

from q , insert in stack)

10
2
8
3

$q:$

14 9 11 5 1 6 10 2 8 3

10
x
8
3

Step 2: (Delete k Elements from stack

to insert in queue)

$q:$

14 11 8 x 3 10 2 8 3 14 9 11 5 1 6

Step 3: (Enqueue q & Dequeue in same q for $N-k$ times)

size of Queue

Another edge

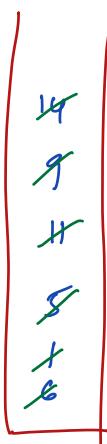
q: 8 8 2 10 14 9 11 5 1 6

→ from rever q

q: 6 X 15 H T H 10 2 8 3

N-k Element

reverse (N-k) Element using Stack



push all Elements Stack to Queue

q: 10 2 8 3 14 9 11 5 1 6

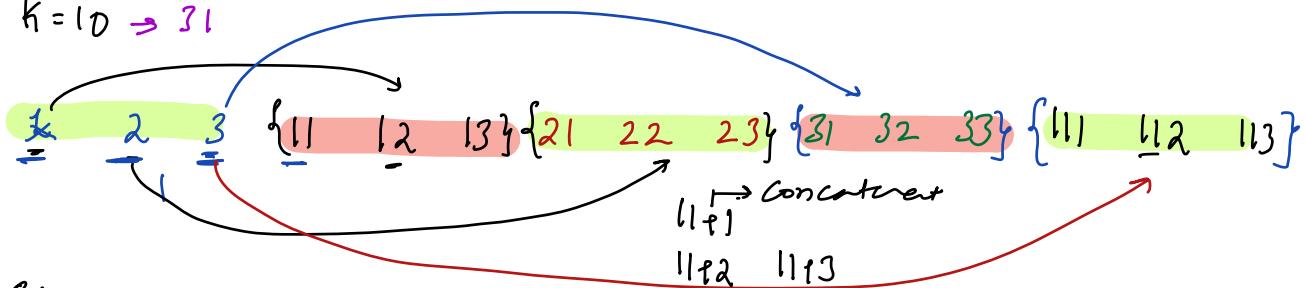
10: 30pm

Q8) Generate k^{th} number in series, by only using digit (1, 4, 2, 9, 3).

$$k=5 \rightarrow 12$$

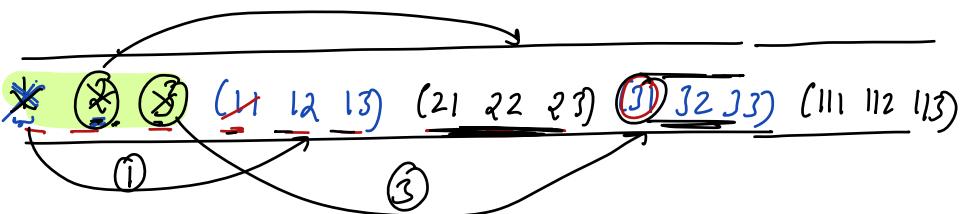
(Increasing order)

$$k = 10 \rightarrow 31$$



Pdca :

Private \mapsto =



At every element

choose if an app fits

an $(1 \bar{q} 2 \bar{q} 3)$

If $y_0 \in (m\ell - m\ell_{-1})$

if (cut == k) {

Labour Element Ans

1

→ We need deck $(k-1)$

Elements from One to

Get k^{th} Element, at front

// In Quer get k^m Element

pdcc:
= 84 cm x 91 t, q

q. `insert(1)` q. `insert(2)` q. `insert(3)`

$$cnt = 3 \Rightarrow$$

$$dcl = 0$$

while (count < k) {

string n = q.front();

q. insert() { $del = del + 1$ }

q. $\lim_{n \rightarrow \infty} (n + 1) \cdot \frac{1}{n^2}$ cut \rightarrow

$$\rightarrow \text{if } (u_4 = k)$$

$$q. \text{ find } (n + \lceil \frac{n}{3} \rceil);$$

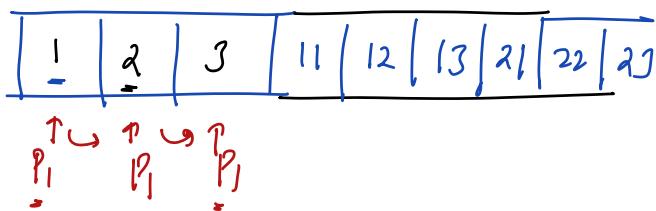
If we need to delete $k-1$ -del no. of

Elements

// Can we do with lrr

lrr < string l

l.add(1) l.add(2) l.add(3)



$p_1 = \emptyset;$

while ($l.size() < k$) {

{ String $x = l[p_1]$

 l.add($x + '1'$)

 l.add($x + '2'$)

 l.add($x + '3'$)

$p_1++;$

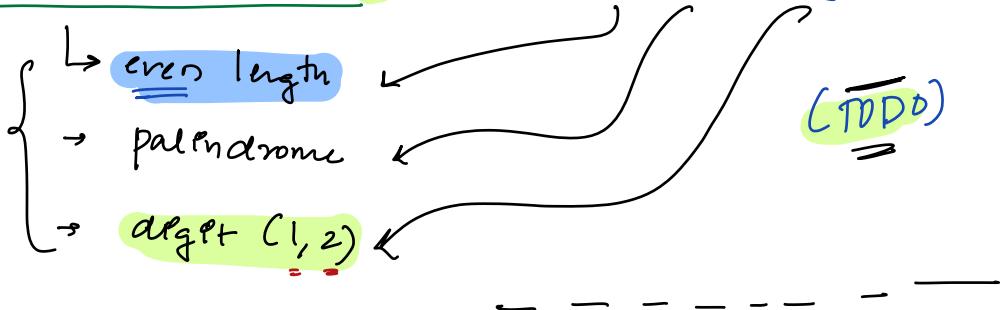
// return $l[k-1]$

Note:

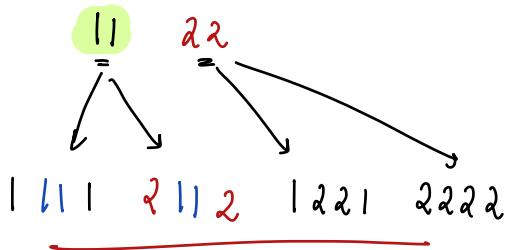
In Queue
we can also
check while
inserting push

$T_C: O(k)$ $S_C: O(k)$

SQ) N^{T} Perfect Number → (When can we say number perfect?)



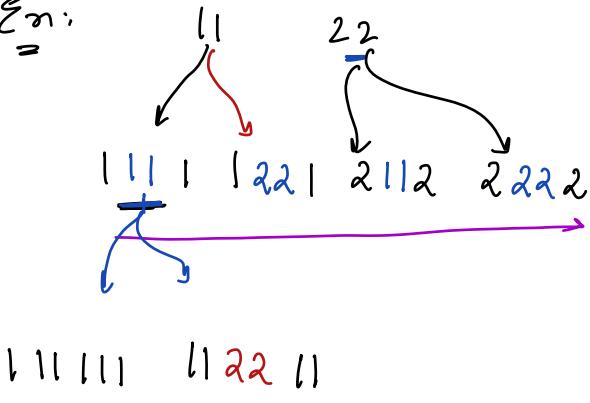
Ex:



} adding same characters
at front and back
→ palindrome ✓

→ order is missing

Ex:



} adding same character
at center, palindrome ✓
→ order is there ✓

Saturday:

- 1) Qu - 2 Stanz: 20 mins
- 2) Degu - 1 Probl
- 3) Infra - Parf
- 4) Portfn Evaluation

Sat: IPL: CSk vs KKR

Build

Dry flat → lcr

Shu

-
- a) 3 session per week
 - b) 6 Session per week

