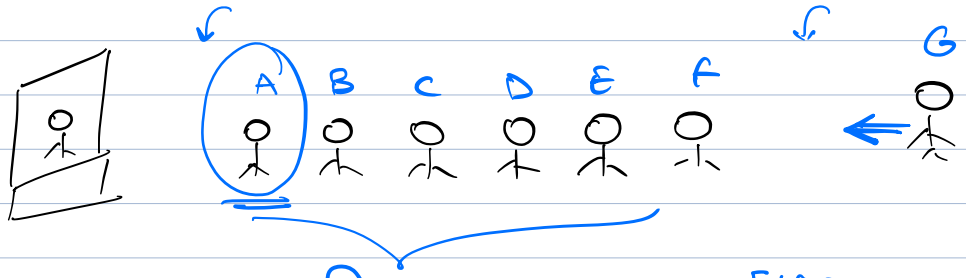


Queue

FIFO
First In First Out

1) Enque(x) → Insert x in the queue

2) Dequeue()/ poll() → Removes the first ele. of queue

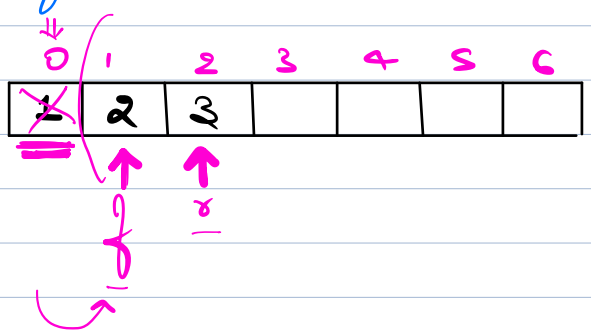
3) front() ⇒ Returns the value of 1st ele of queue

4) isEmpty() ⇒ Return True if queue is Empty else false

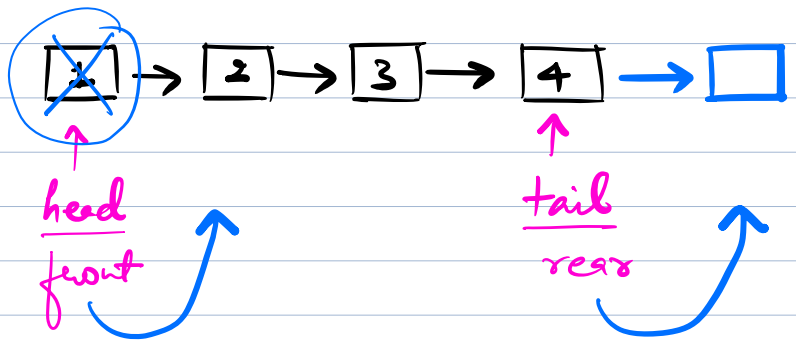
T.C. = O(1)

Implementation

1) Arrays.



2) Linked List



Q) Implement a queue using only stacks

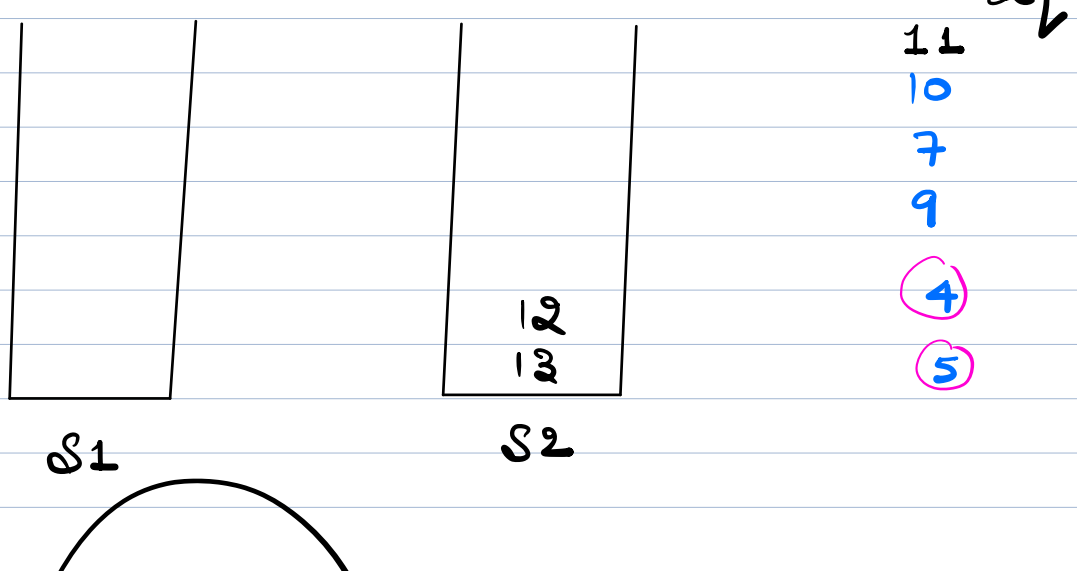
Enqueue(x) > O(1)

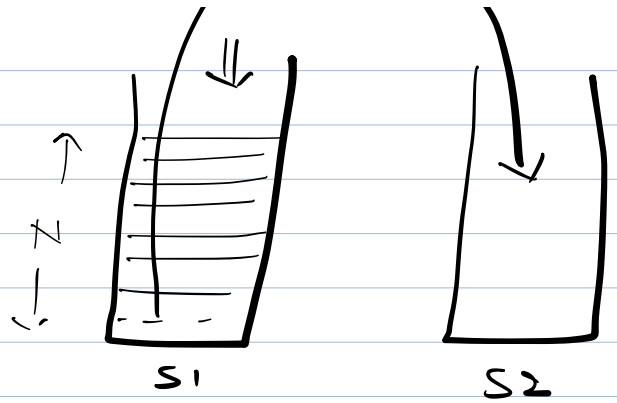
Dequeue()

push
pop
top
is Empty

5, 4, 9, 7, 10, dq, 11, 12, 13, dq, dq, dq

dq





1 st deq = O(N)

(N-1) deq = 1 × (N-1)

N deques = 2N-1 ≈ 2N

1 deque = O(2) = O(1)

Amortized

Q

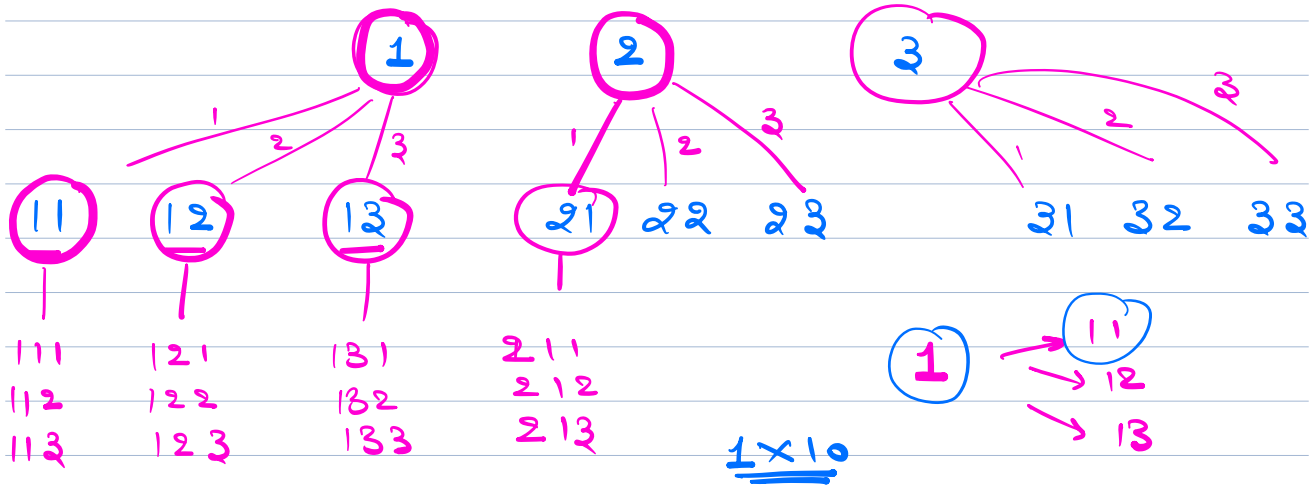
3 digits ⇒ {1, 2, 3}

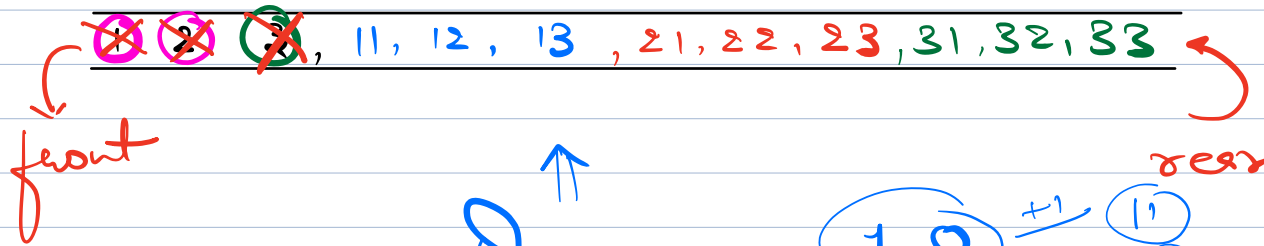
Given N, return the Nth no. formed using these digits.

N = 8

1 → 1
2 → 2
3 → 3
11 → 4
12 → 5
13 → 6
21 → 7
22 → 8
23
31
32
33
111
112
113
⋮

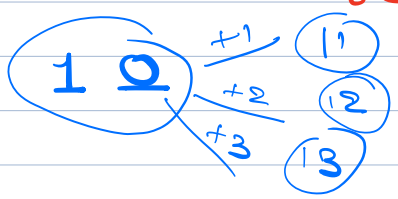
Ans ← 22





Queue

T.C. = O(N)

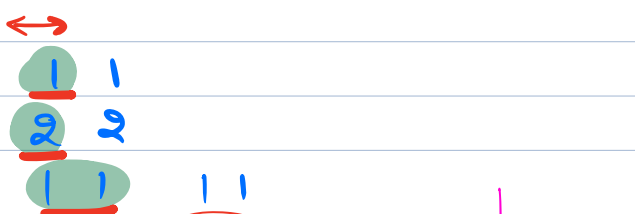


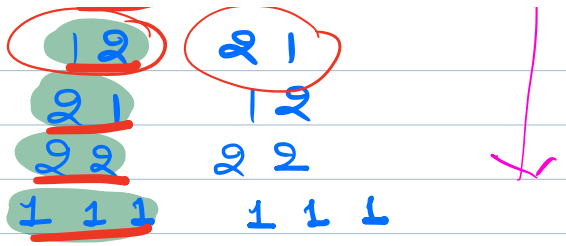
Q

Find the Nth perfect no.

- Digits : {1, 2}
- Even length
- palindrome.

1, 2, 11, 12, 21, 22, 111, 112, 121, 122,
211, 212, 221, 222, 1111, 1112 ....





Q

Given a queue.

Reverse the queue.

2, 3, 5, 9, 1, 8, 7

↓

7, 8, 1, 9, 5, 3, 2

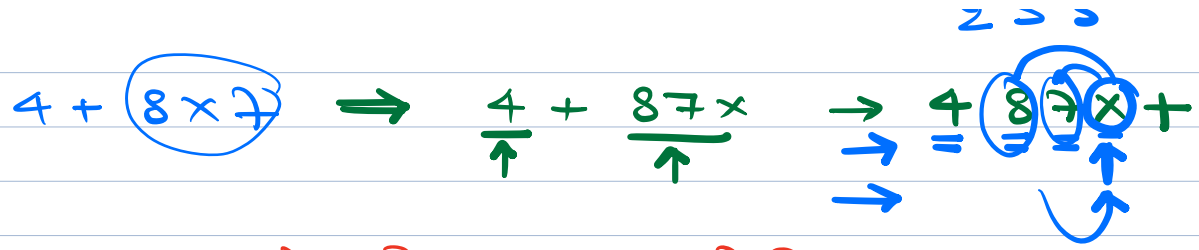
⇒ Dequeue the entire queue in a stack

⇒ Enqueue the stack back.

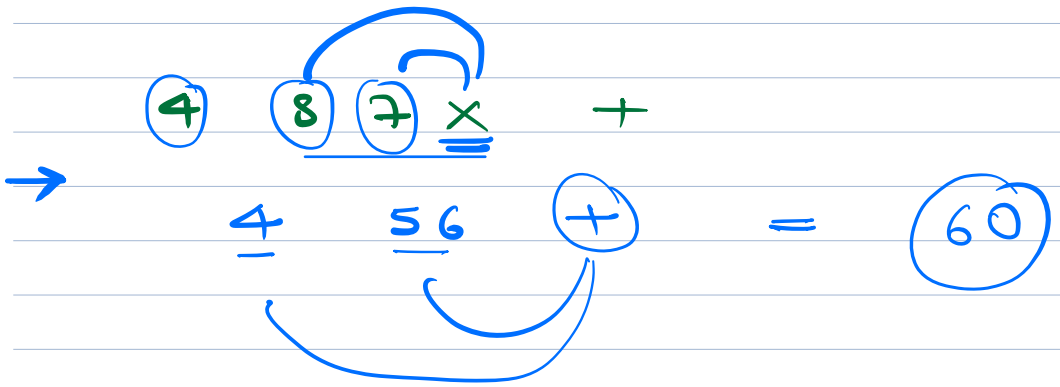
T.C. = O(N)

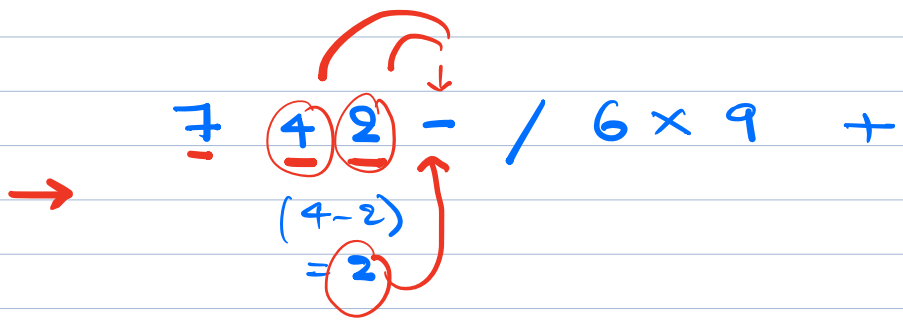
S.C. = O(N)

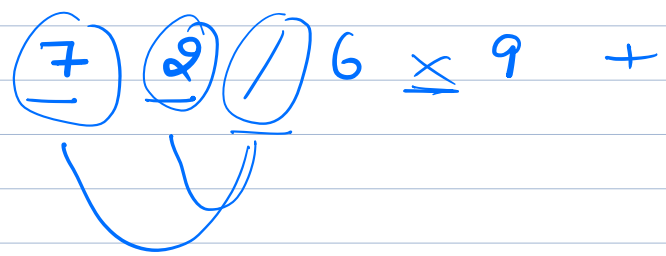
BODMAS

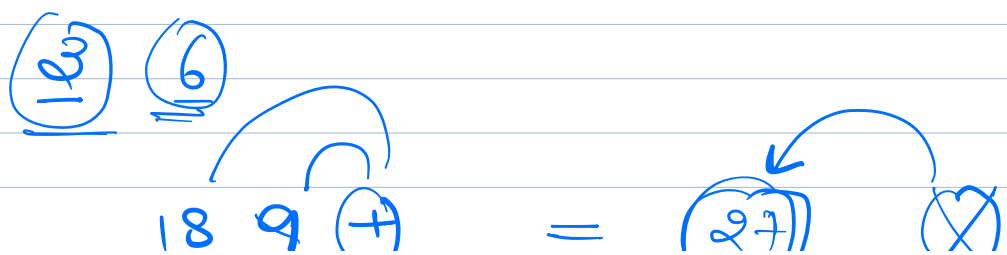


A × B → A B ×
A ÷ B → A B ÷
A + B → A B +
A − B → A B −





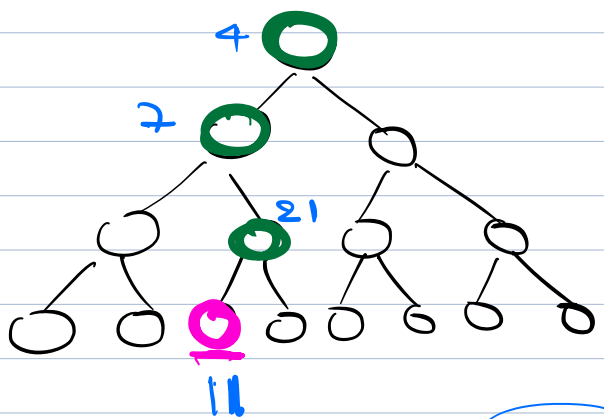


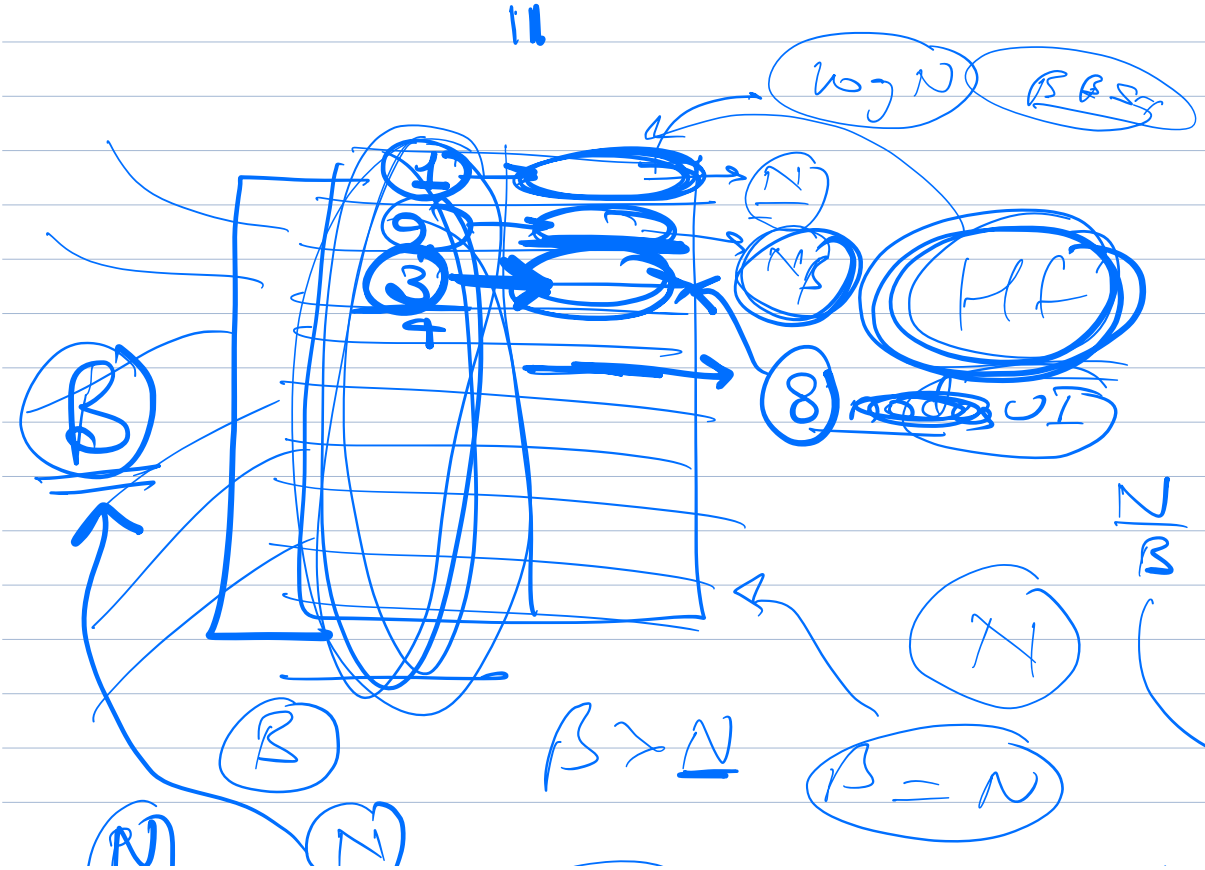


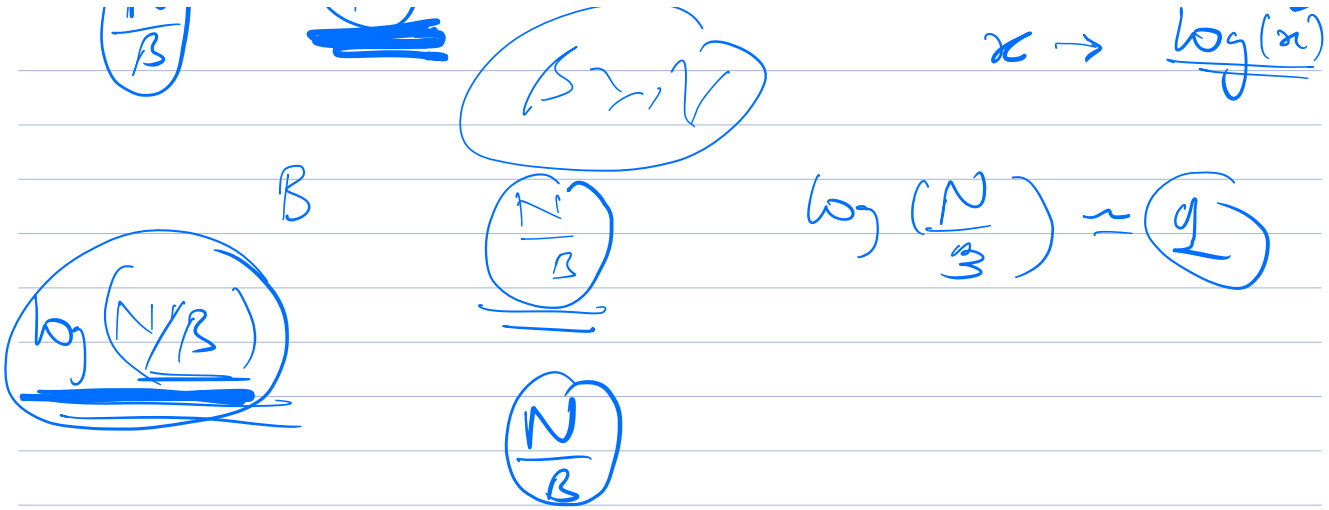
⇒ Stack

H.W: Code

Doubt







Code

```
Queue<INT> Q

N<=3 { return N }

count = 3

while (count < N) {

    int x = q.dequeue();

    for (i = 1; i <= 3; i++) {

        int next = x*10 + i
        count++;

        if (count == N) {
            return next;
        }
        q.enqueue(next);
    }
```

b