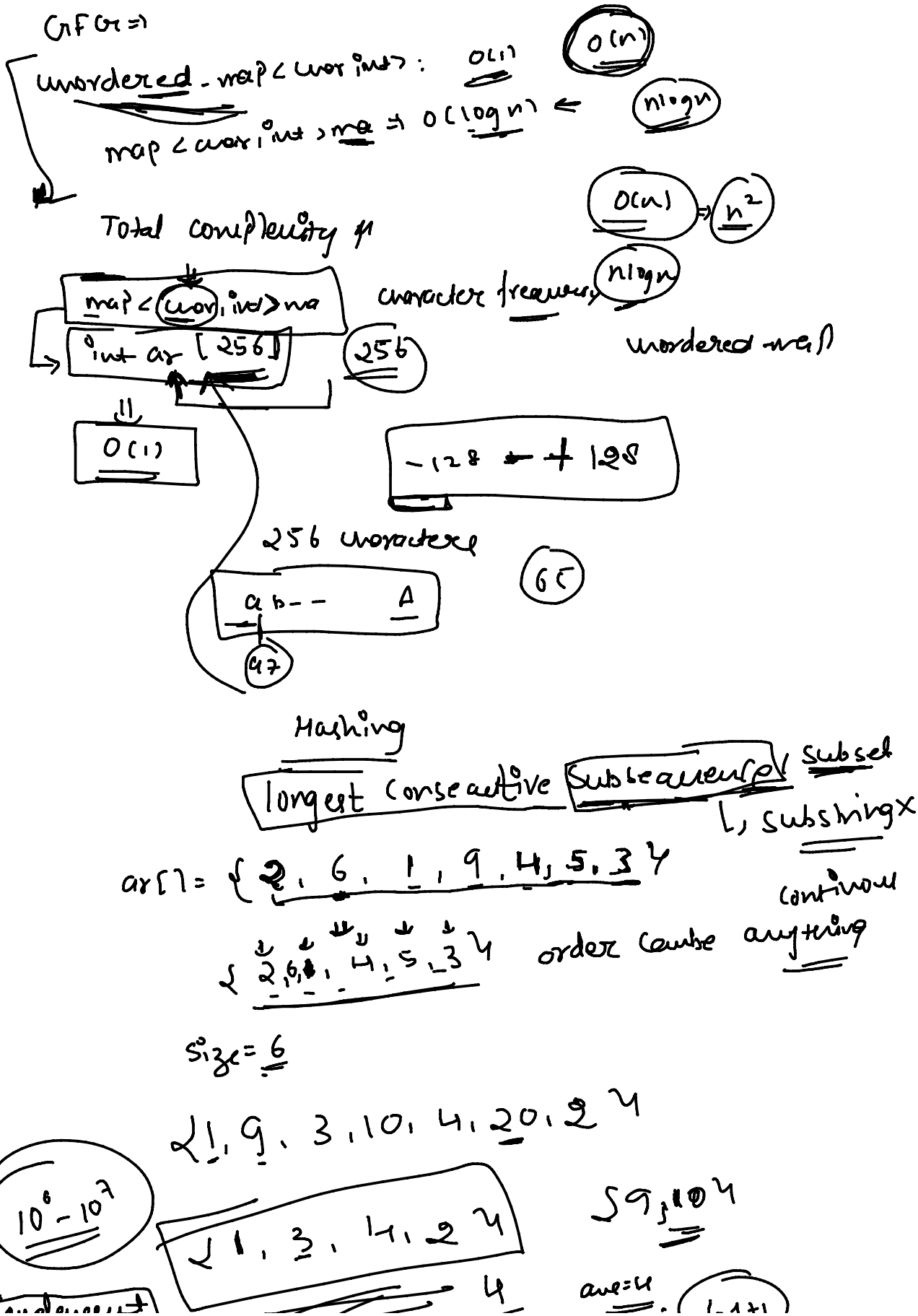
Class 23
02 March 2023 20:56

Class-23



Maxelement
10^{18}

4   ans=4   i   i=1+1

T.C = O(n)

Brute Force → array[20] = {1 1 1 1 ... 1 1 ... 14

keyvalue pair   Memory waste   array Out   max consecutive one

Subsequences generate, check each subsequence

T.C = O(exponential)

{1, 2, 3, 10^6} => arr[10^6]   waste

Set<int> s

{2, 9, 3, 10, 4, 20, 1}

Element Present in the given array

mp.
- 1 → 1
- 9 → 1
- 3 → 1
- 10 → 1
- 4 → 1
- 20 → 1
- 2 → 1

Consecutive Subsequence

1, 2, 3, 4, 5, 6 - - -

2   1   {1, 2, 3, 4}

mp[3] = 1   mp[4], mp[5] X

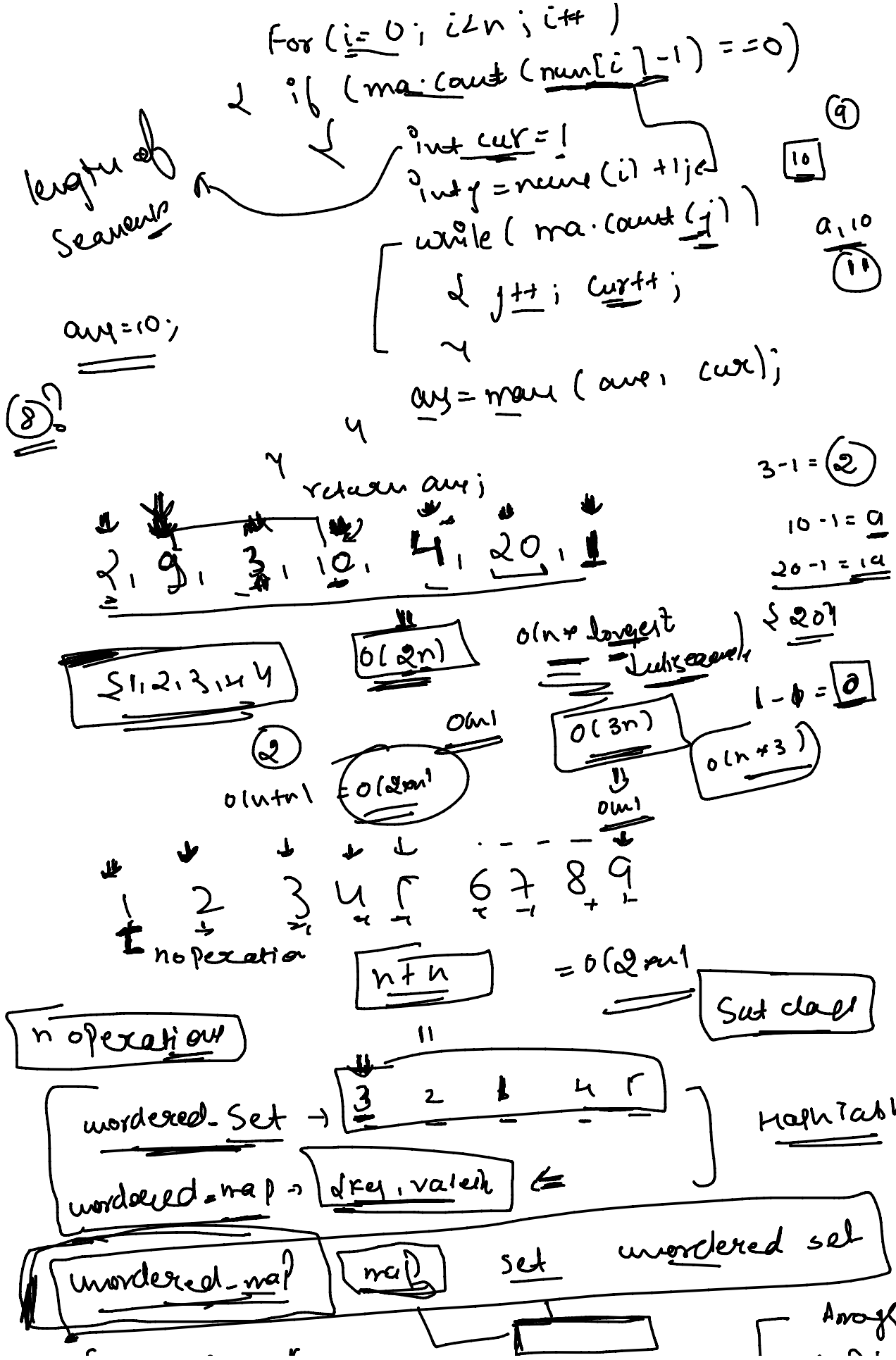
2, 3, 4 => {2, 3, 4} {9, 10} {10}

{3, 4} {4} {20}

unordered_map<int, int> mp;

```
for (auto i : nums)
        mp[i] = 1;

int ans = 0;
for (i = 0; i < n; i++)
    if (mp[nums[i] - 1] == 0)
```

mp[i]


For (i=0; i<n; i++)
if (ma.count(nums[i]-1)==0)
int cur=1
int y=nums[i]+1
while (ma.count(y))
y++; cur++;
ans=max(ans, cur);
return ans;
ans=0;
length of sequence
2, 9, 3, 10, 4, 20, 1
{1,2,3,4}
O(2n)
O(n * longest subsequence)
O(3n)
O(n+3)
O(n+n)
O(2n)
n operation
unordered_set → 3 2 1 4
unordered_map → key, value
unordered_map
map
set
unordered set
HashTable
Array

Elements Track
↳ Sliding window

functions
functional Pointers
lambda expression

custom comparator

custom Hashing

Arrays
2 Pointers
Binary search
Hashing

Data type Hash CIX

custom Hash function

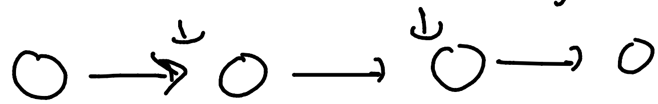
Stacks, Queues, Deque → sliding window

window max element

- Arrays
- HashTable
- Linkedlist
- Trees
- Graphs

Data structure features
→ elements insert (end_O(1))
Random Access

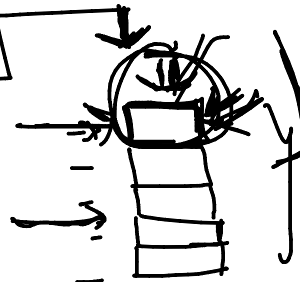
→ element Randomly store internally



Array

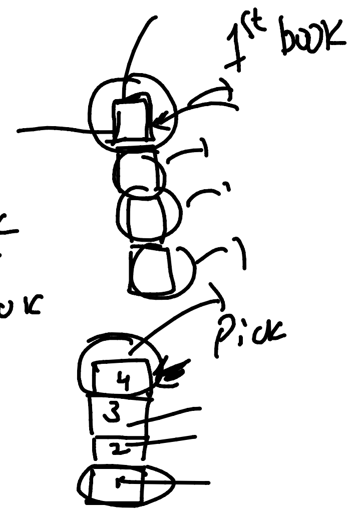
insertion easy

Stack



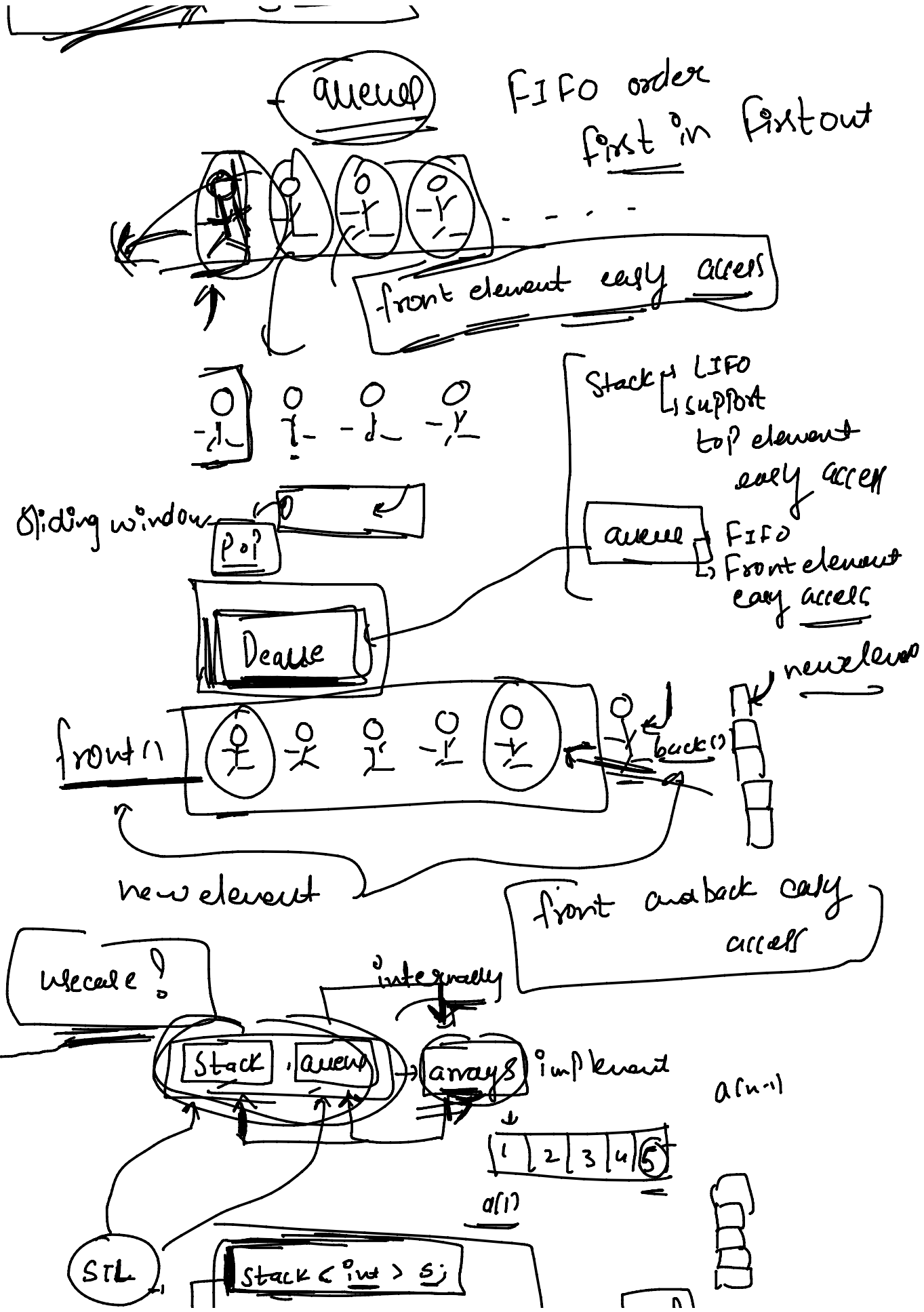
pile of book
stack of book

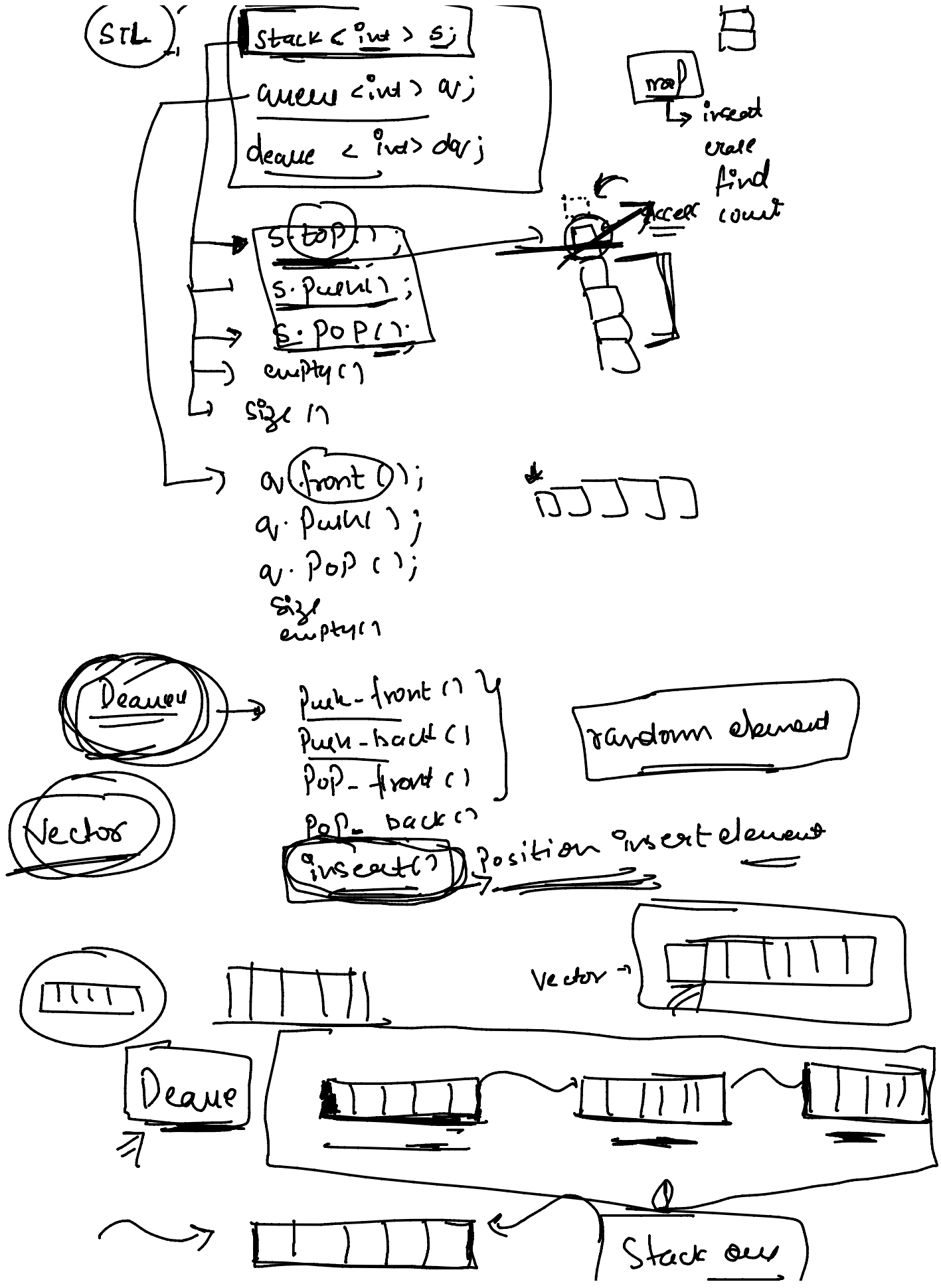
Easy access



LIFO order

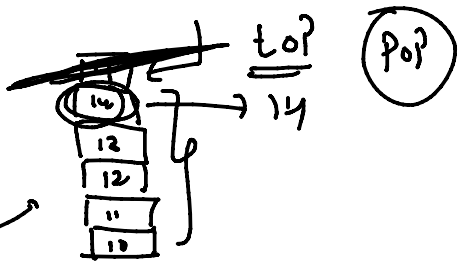
Top element Easy access

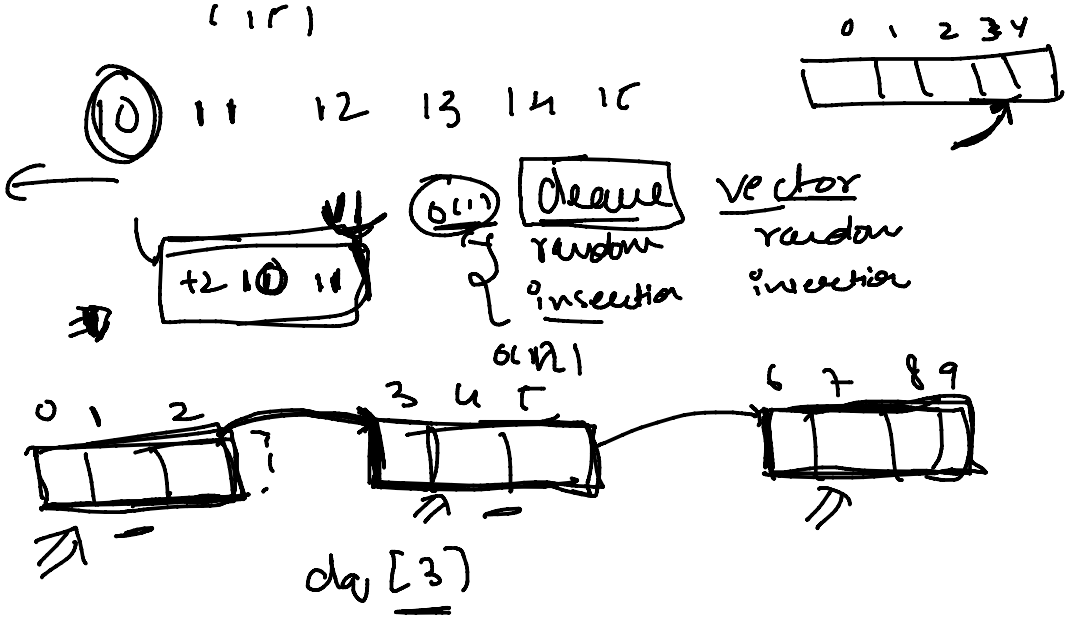

queue
FIFO order
first in firstout
front element easy access
Stack LIFO
support
top element
easy access
queue FIFO
Front element
easy access
Sliding window
Deque
front()
back()
new element
new element
front and back easy
access
integrally
Stack
queue
arrays
implement
a[n-1]
a[1]
STL
Stack<int> s;


STL
stack<int> s;
queue<int> q;
deque<int> dq;
s.top();
s.push();
s.pop();
empty()
size()
q.front();
q.push();
q.pop();
size
empty()
map
insert
erase
find
count
access
Deque
Vector
push_front()
push_back()
pop_front()
pop_back()
insert()
position
insert element
random element
Vector
Deque
Stack

Deque function

```
Stack <int> S;
S.push(10)
       (11)
       (12)
       (13)
       (14)
       (15)
```





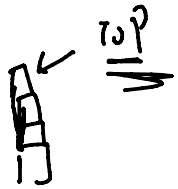
Deque bole large size compared to vector

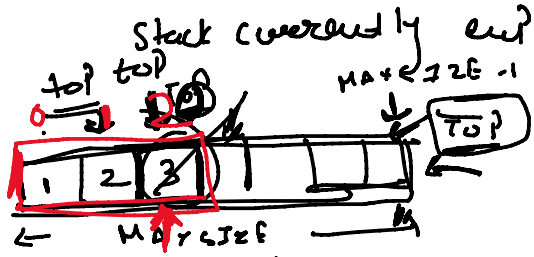
Stack, queue, deque ↳ STL

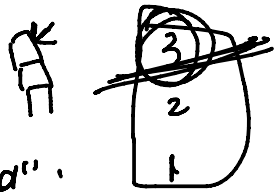
Implementation

How to implement Stack using Array;

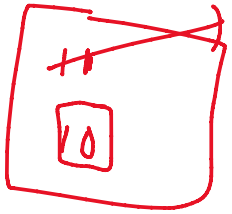
```
#define MAXSIZE 100

class Stack {
    int top;
  Public:
    int ar [MAX SIZE];
    Stack()                    no element
    { top = -1;
    }
    bool push (int n)
    {   if (top >= (MAXSIZE -1))
        {  cout << " False element can't be pushed";
           return 0;
        }
        else // array has space for new element
        {   top++;
            ar[top] = n;
            cout << " Element has been inserted";
            return 1;
        }
    }
    int pop()                  Top element of stack remove it
    {   if (top < 0)           No element in Array
        {  cout << " Underflow ";
           return -1;
```

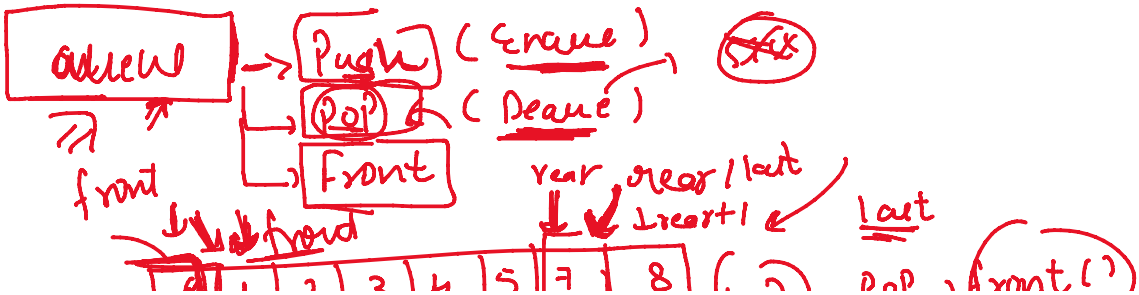


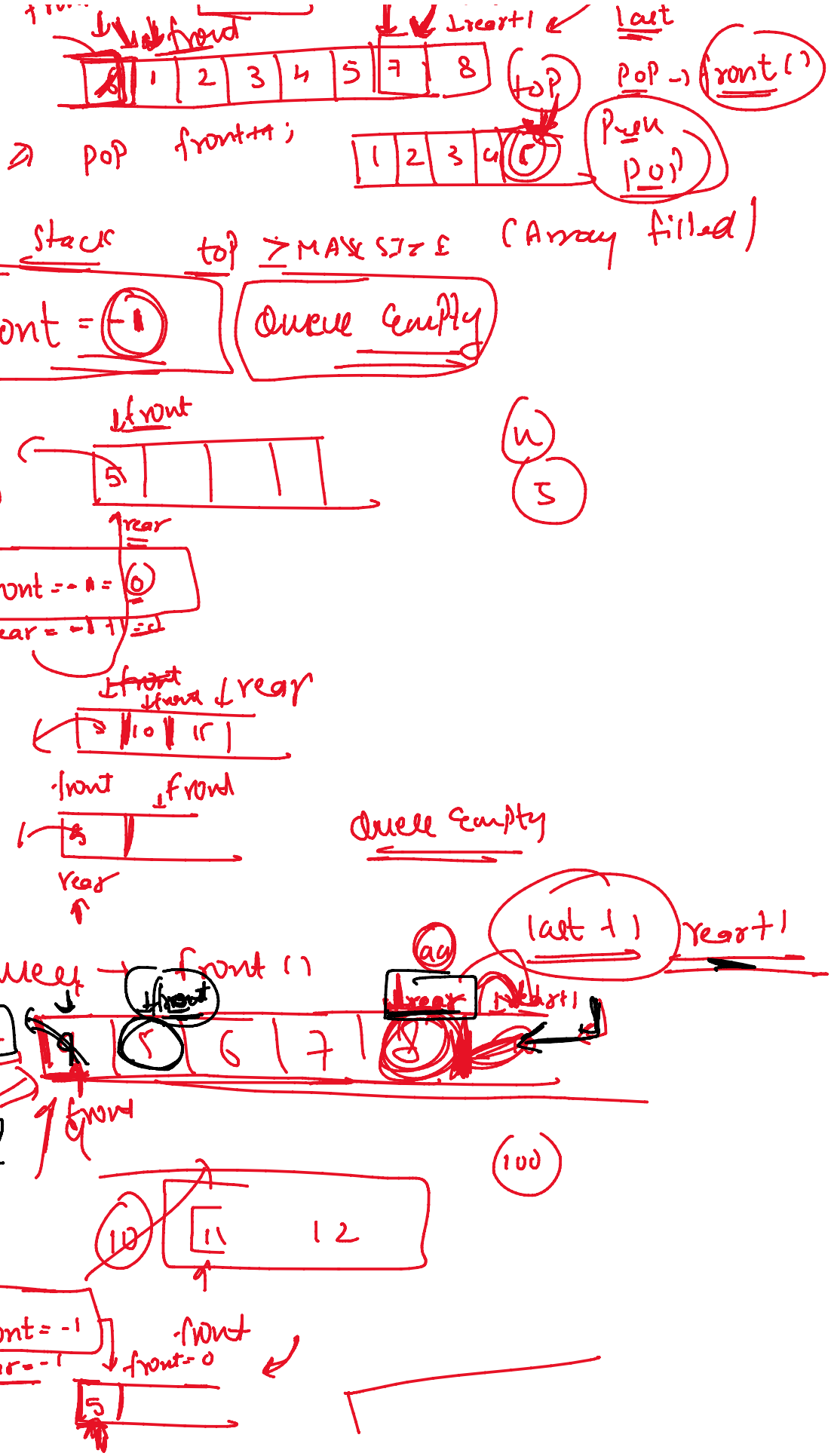


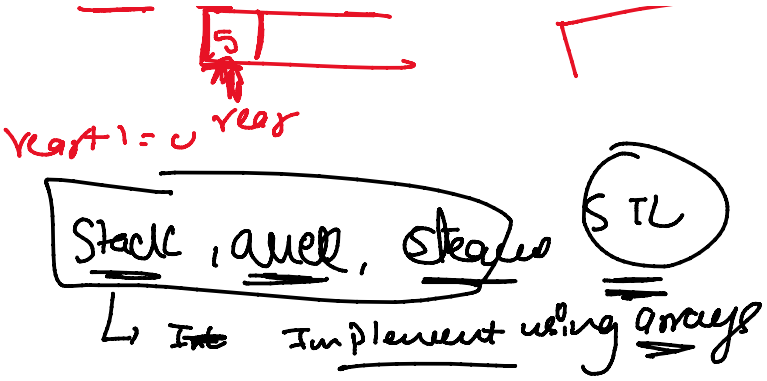


```
    }
    else
    {   int x= a[top];
        top--;
        return x;
    }
}
int topAccess()
{   if(top<0)
    { cout<<"stack is empty";
    }
    return a[top];
}
bool isEmpty()
    {  if(top<0)
        return true;
       return false;
    }
int size(){
    return top+1;
    }
};
```






front
rear+1
last
1 2 3 4 5 7 8
top
pop -> front()
pop front++;
1 2 3 4
push
pop
Stack
top ≥ MAX SIZE (Array filled)
front = -1
Queue empty
front
pop
5
rear
front = -1 = 0
rear = -1 + 1 = 0
front
rear
10 15
front
rear
Queue empty
last + 1
rear+1
queue
front()
front
5 6 7 8
front
a[front]
100
10
11 12
front = -1
rear = -1
front = 0
5


5
rear+1 = 0
rear
Stack, queue, deque
STL
Implement using arrays