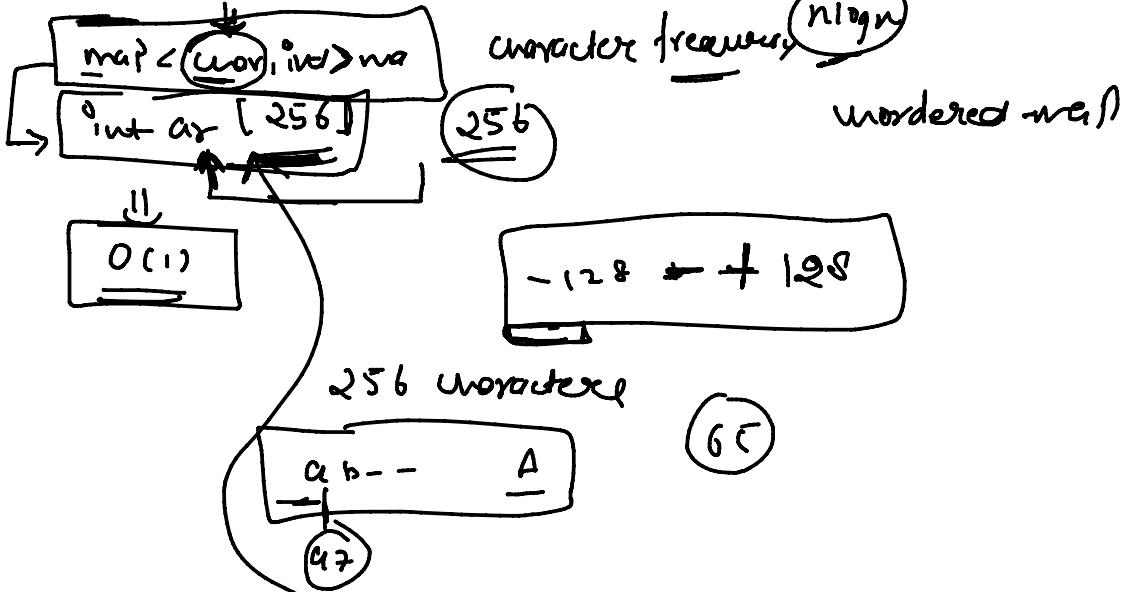


Class - 23

GFG or =

unordered\_map < (char, int) :  $O(n)$

map < (char, int) :  $O(\log n)$   $\leftarrow n \log n$

Total complexity  $\pi$ 

Hashing

longest consecutive Subsequence  $\leftarrow$  subset, substring

$arr[] = \{2, 6, 1, 9, 4, 5, 3\}$

$\{2, 6, 1, 4, 5, 3\}$  order can be anything

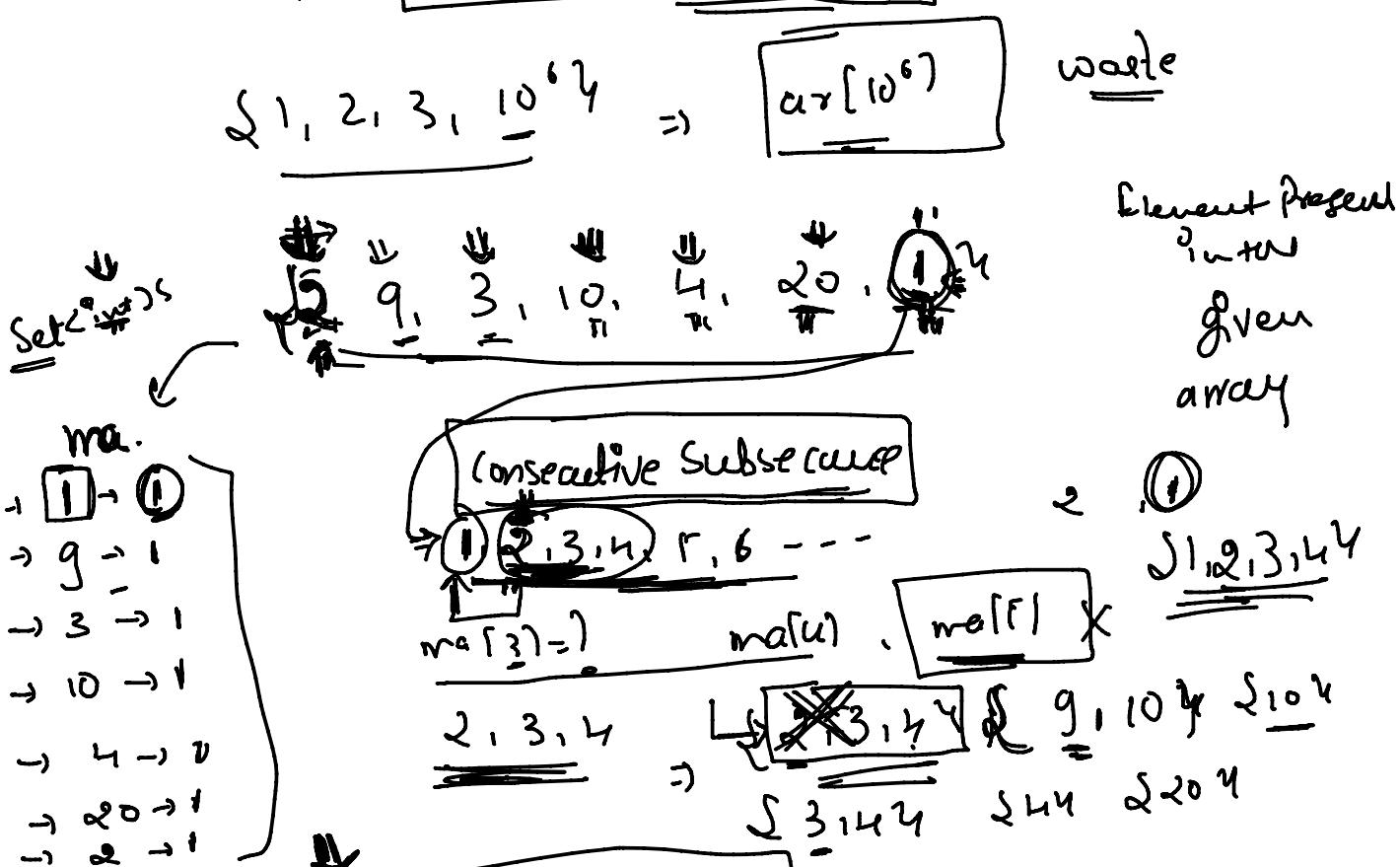
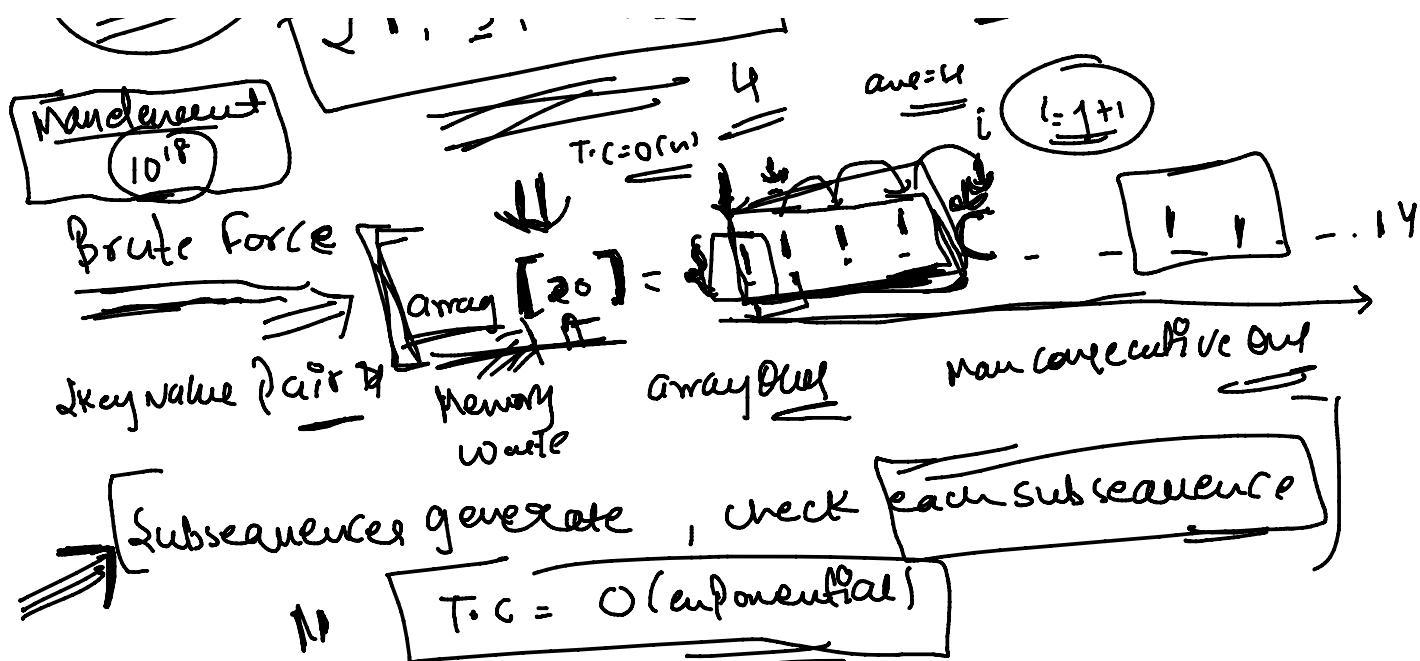
Size = 6

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

 ~~$10^6 - 10^7$~~ 

$\{1, 3, 11, 2\}$

 $59, 10^4$ ave = 4.  $\frac{1+1+1}{3}$



unordered\_map<int, int> ma;

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

int ave = 0;

```
for (i = 0; i < n; i++)
    if (ma[i] == 0)
```

ma[i]

For ( $i = 0$ ;  $i < n$ ;  $i++$ )  
 2 if ( $ma \cdot \text{count}(\underline{\text{num}[i] - 1}) == 0$ )

int cur = 1  
 int j = new(i) + 1  
 while ( $ma \cdot \text{count}(\underline{j})$ )  
 2  $j++$ ; cur++  
ans = max (ans, cur);

ans = max (ans, cur);

return ans;

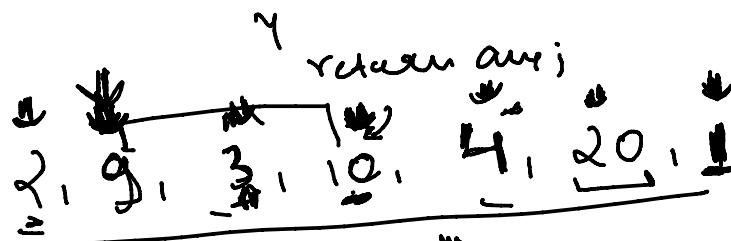
(8) length of Seams

ans = 10;

(9)

(10)

(11)



$$3-1 = 2$$

$$10-1 = 9$$

$$20-1 = 19$$

{1, 2, 3, 10, 4}

O(2n)

O(n + longest seam)  
= length of seams

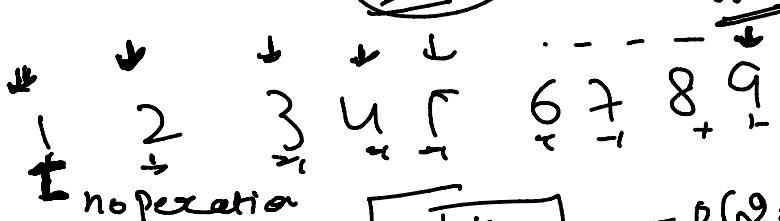
20

O(n)

O(n)

$$1-0 = 0$$

O(n+3)



n+n

= O(2n)

Set class

n operations

unordered\_set →

3 2 1 4 5

MapTable

unordered\_map →

key, value

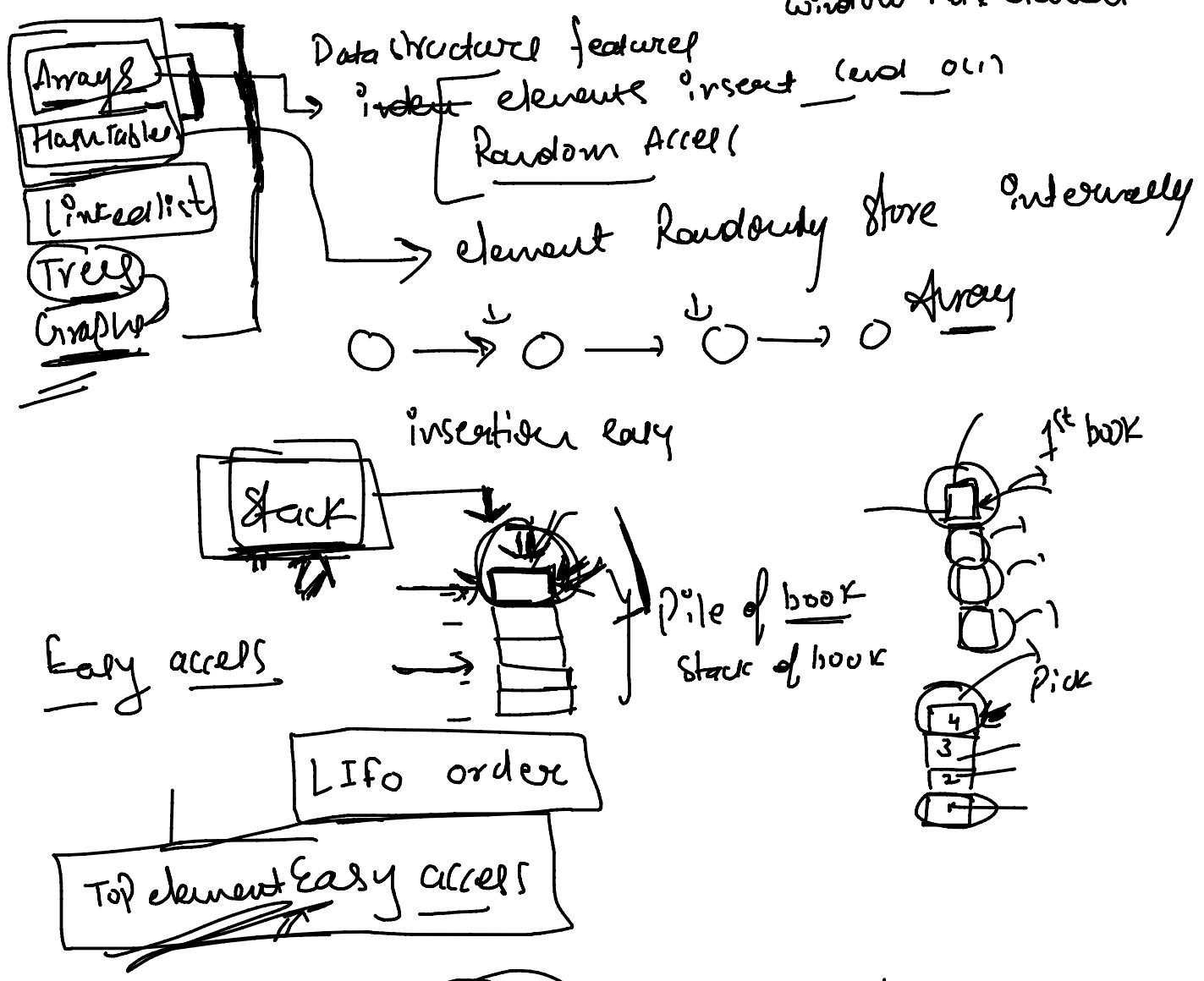
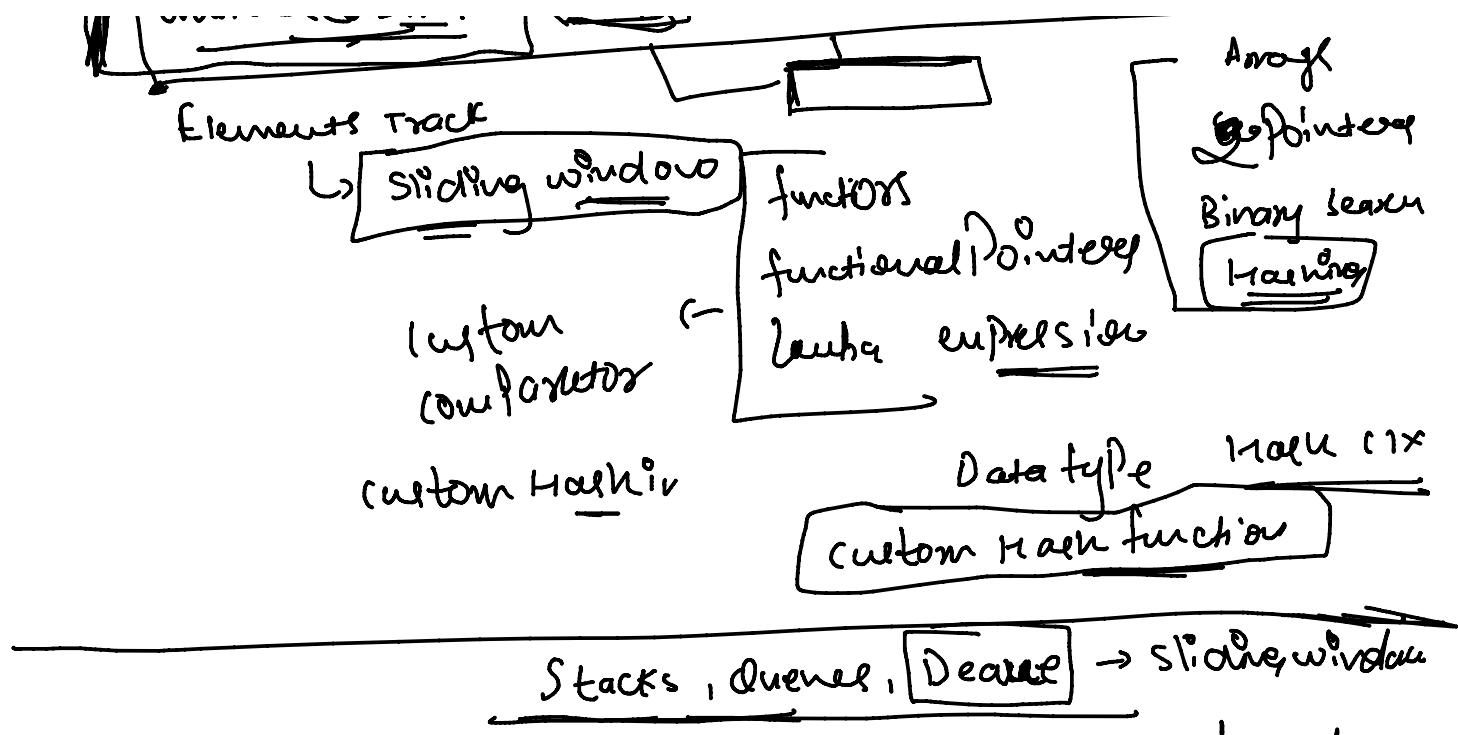
unordered\_map?

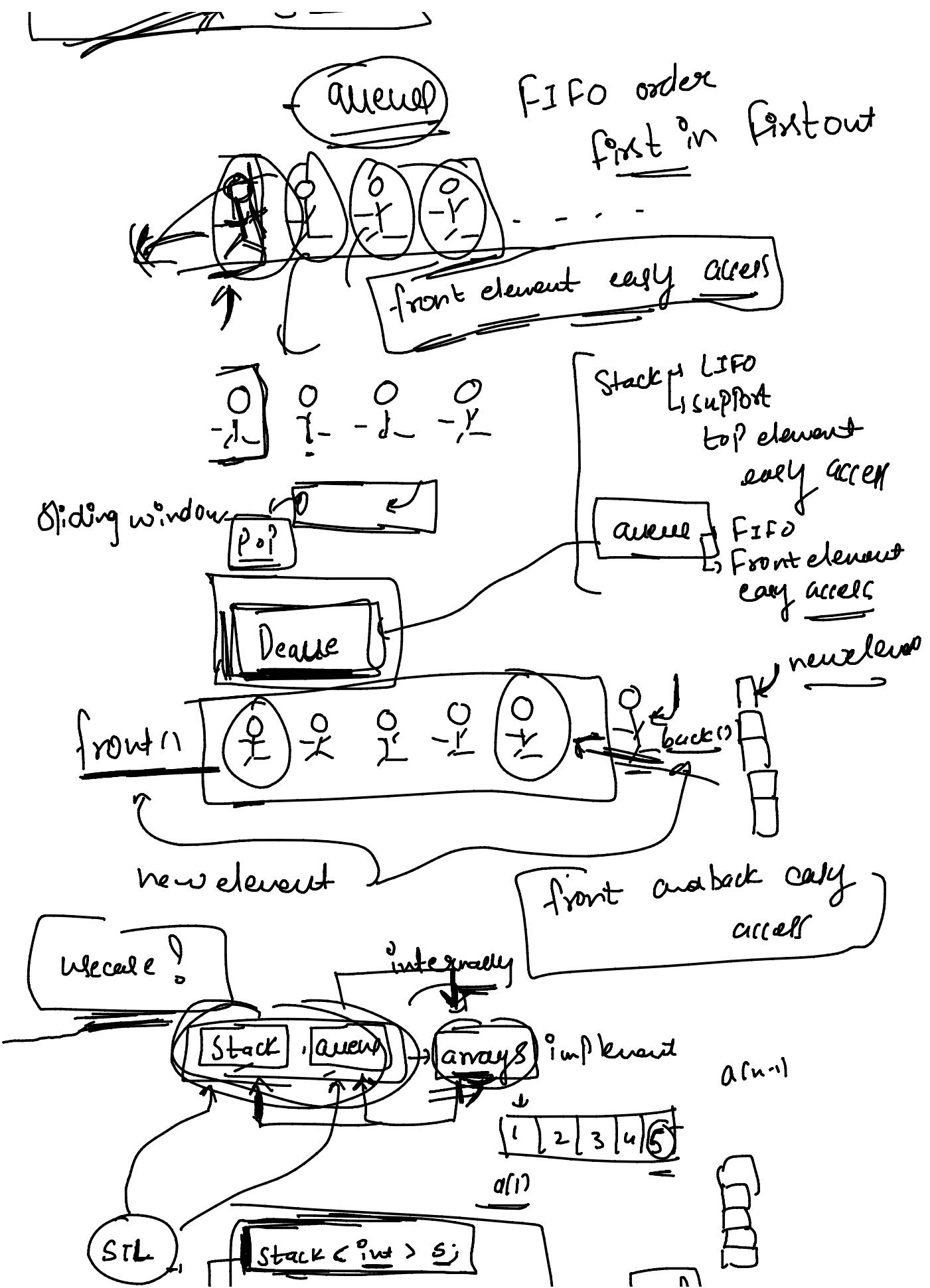
map

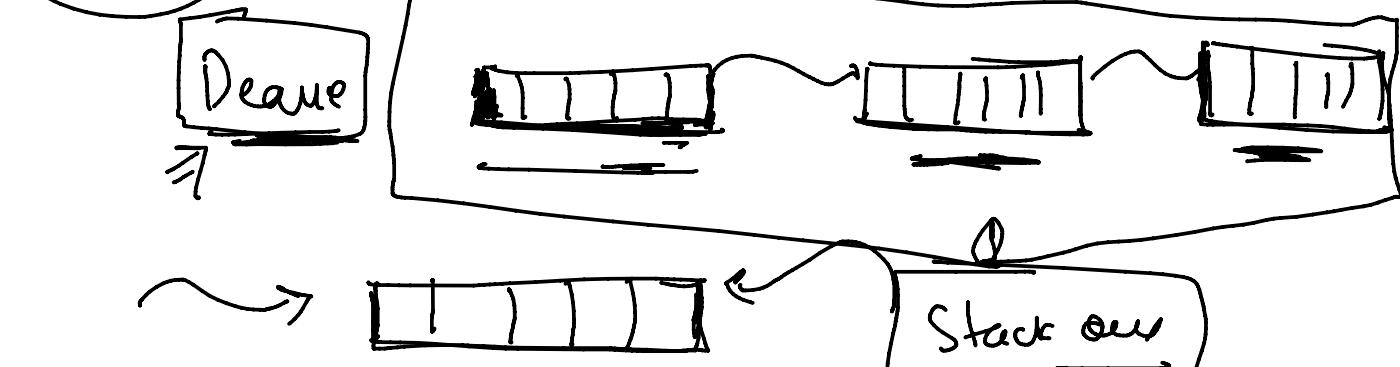
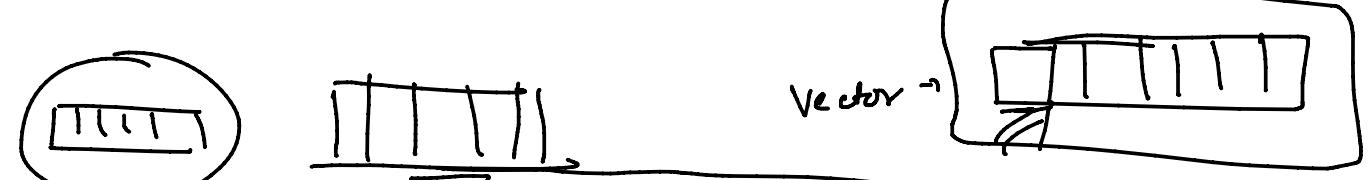
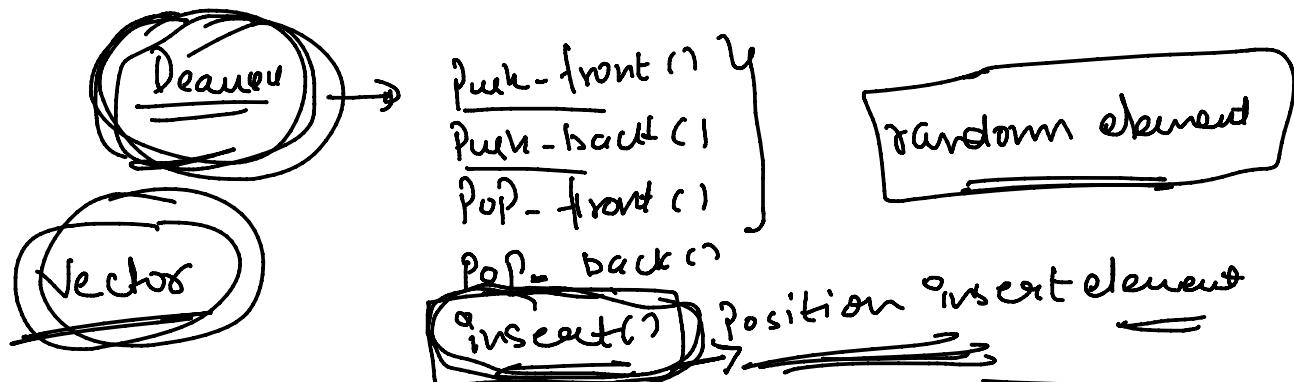
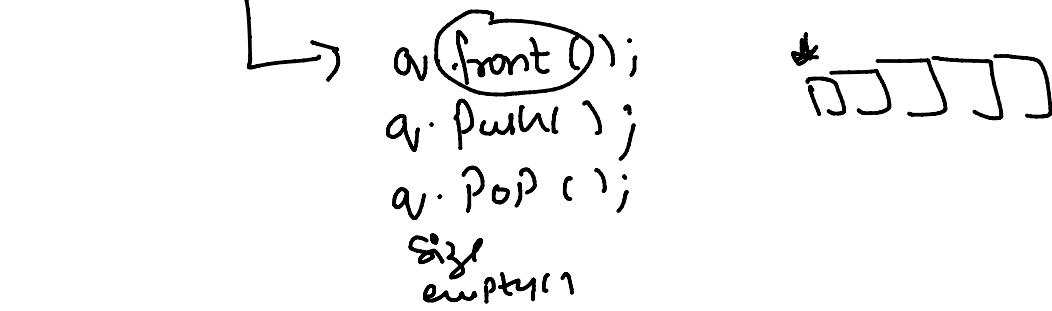
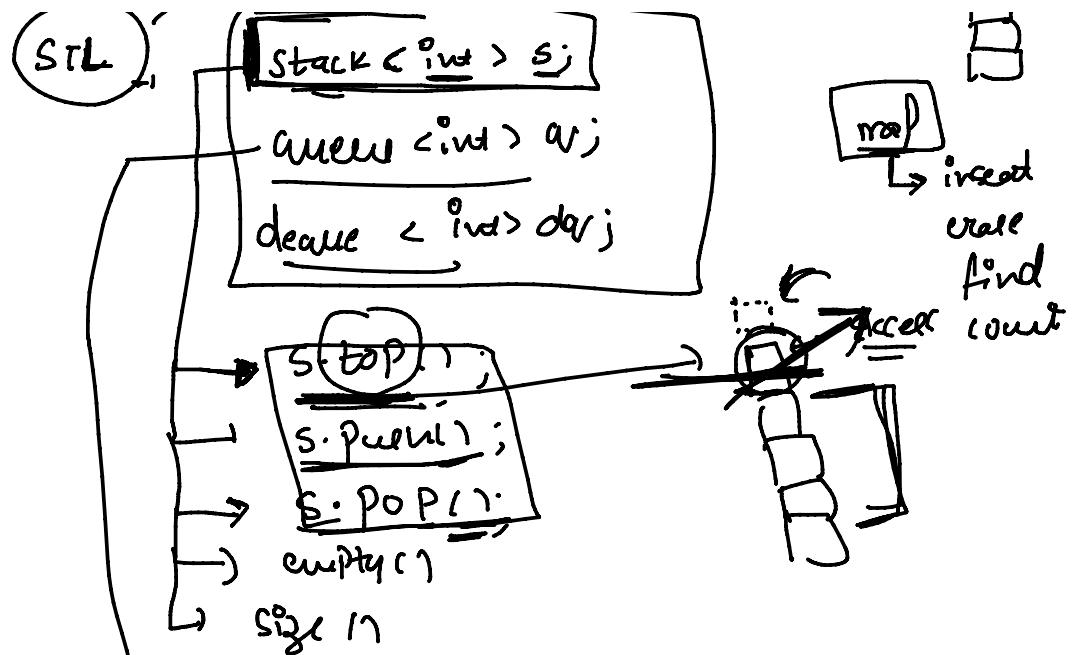
set

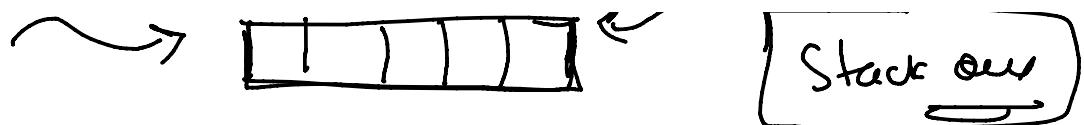
unordered set

Analog









## Deque function

Stack <int> s;

s.push(10)

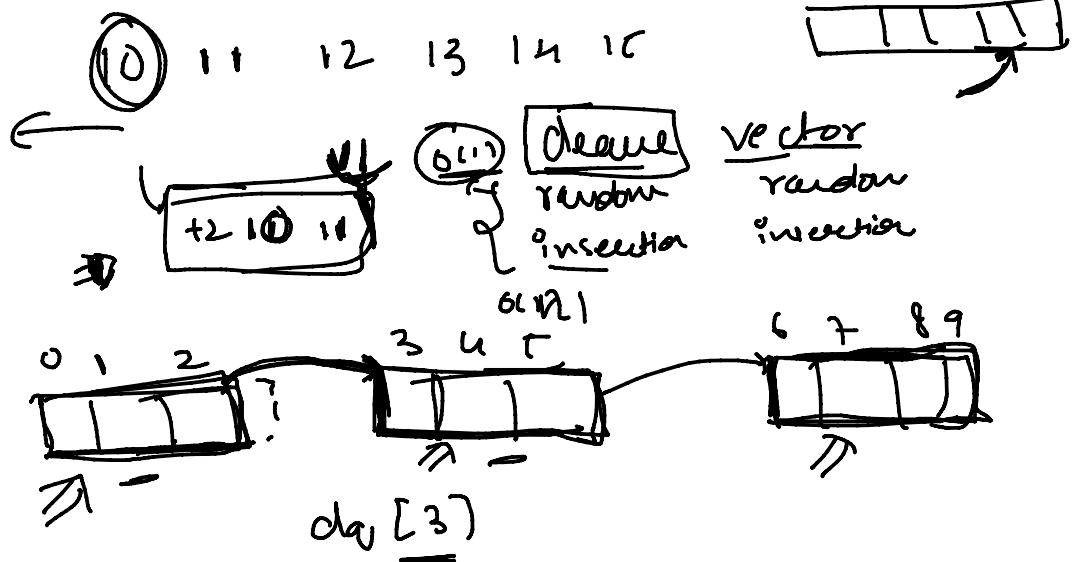
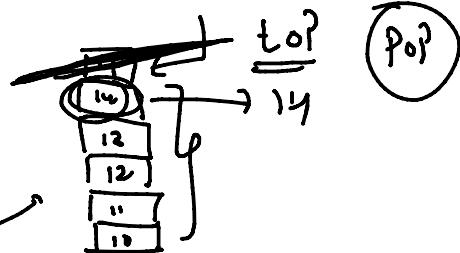
(11)

(12)

(13)

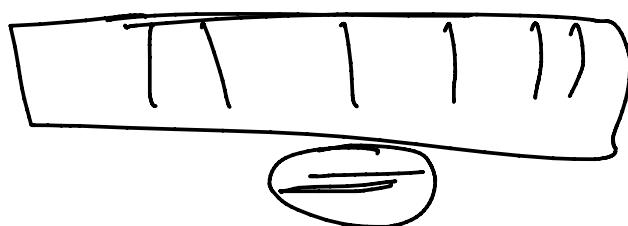
(14)

(15)



deque bde large size compared to vector

vector ->



stack, queue, deque VS SLL

Implementation

How to implement Stack using Array ;

```
#define MAXSIZE 100
```

```
class Stack {
```

```
    int top;
```

```
public:
```

```
    int ar [MAXSIZE];
```

```
Stack() no element
```

```
{ top = -1; }
```

```
}
```

```
bool push (int n)
```

```
{ if (top >= (MAXSIZE - 1))
```

cout "false element can't be pushed";

```
return 0;
```

```
}
```

else if array has Space for new element

```
{ top++;
```

```
ar [top] = n;
```

cout "Element has been inserted";

```
return 1;
```

```
}
```

```
y
```

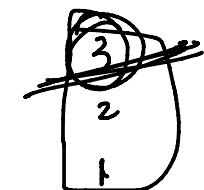
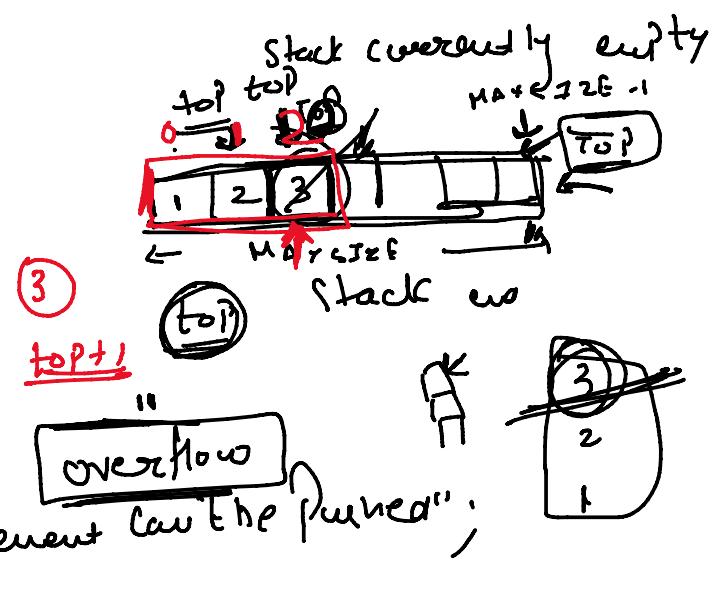
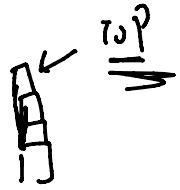
```
int pop()
```

top element of stack remove it

```
{ if (top < 0) No element in Array
```

cout "Underflow";

```
return -1;
```



```

    ~
    {
        int u = a[ top? ];
        top--;
        return u;
    }

```

```

    ~
    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;
}

```

4;

