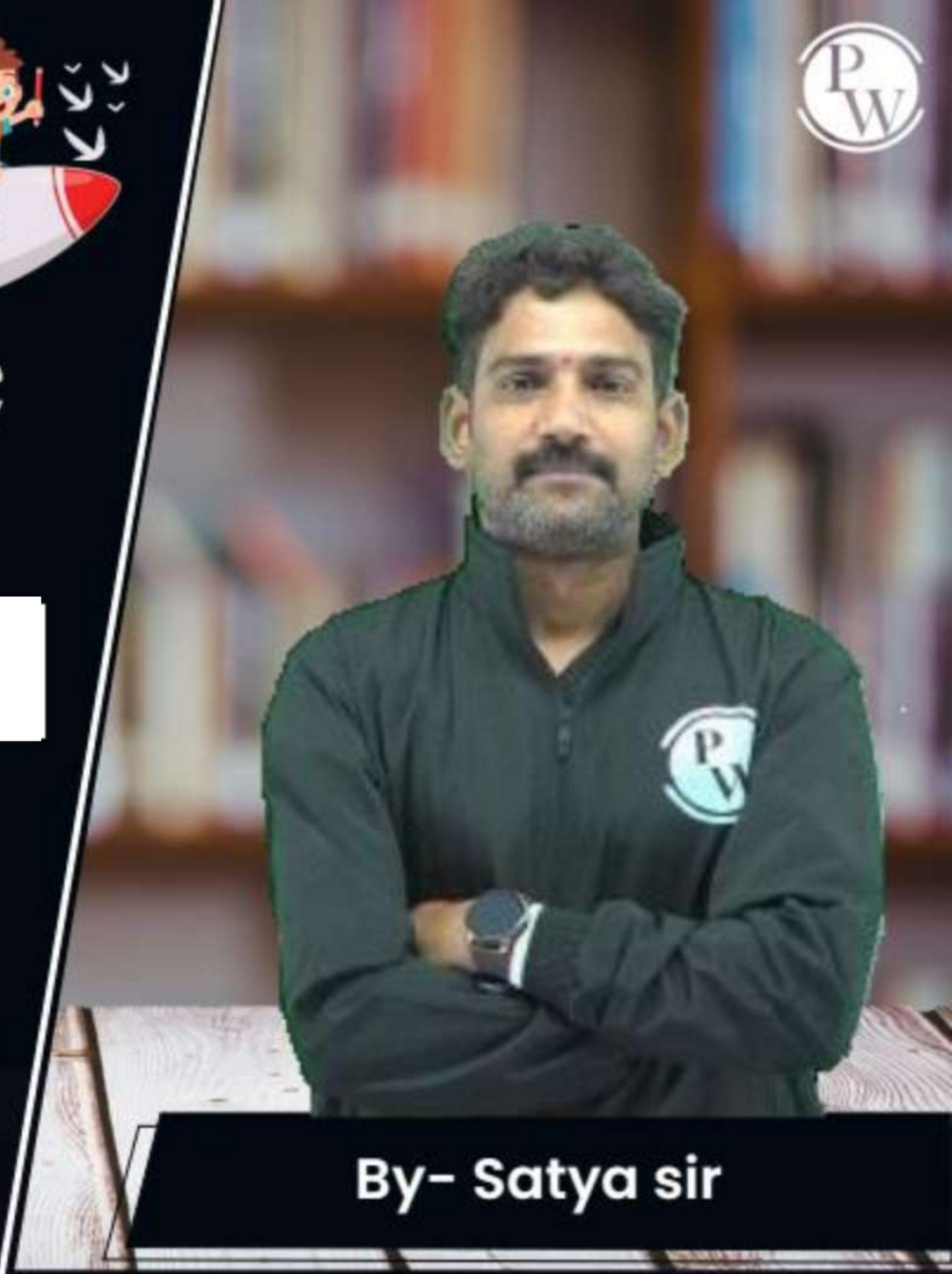
Data Science & Artificial Intelligence

Data Structure through Python

Super 1500+

Lecture No.- 03



By- Satya sir

Topics to be Covered

- Applications of stack
- Permutations
- Operation on stack

Last class

- Queues
- Hash Tables.

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 1

#Q. The Postfix Expression for an infix expression a+b*(c^d-e)^(f+g*h)-i is

A. abcd^e-fgh*+^*+i-

B. abcd^e*-fgh*+^+i-

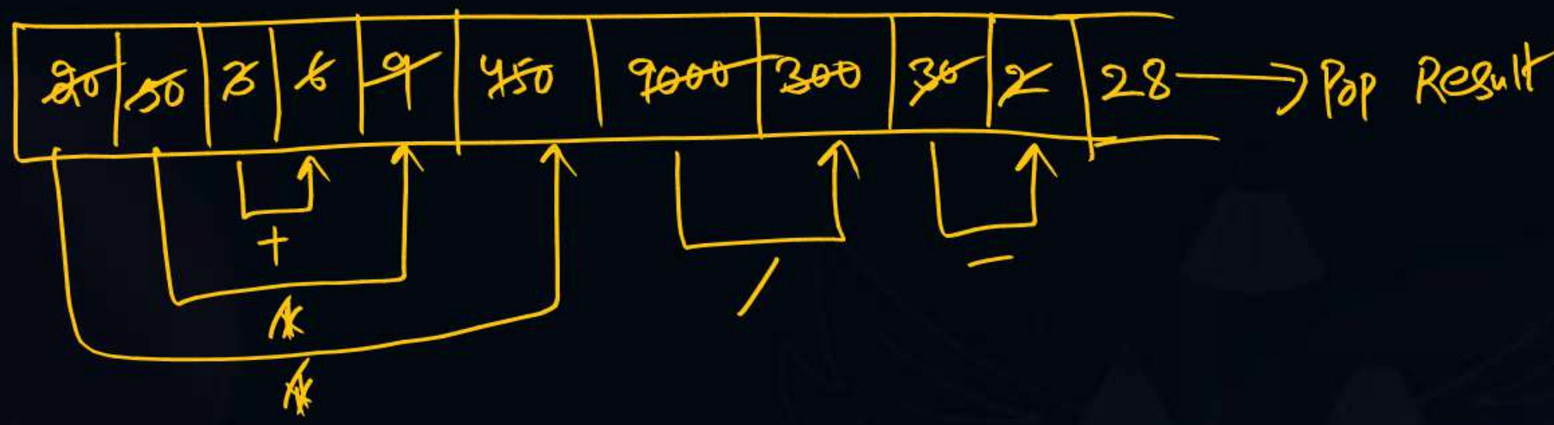
C. abcd^e-*fgh+^*+i-

D. abcd^e-fgh+^**+i-

Postfix: a b c d ^ e - f g h * + ^ * + i -

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 2

#Q. The result after evaluating the postfix Expression 20 50 3 6 + * * 300 / 2 – is 28



SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 3

#Q. The following postfix expression with single digit operands is evaluated using a stack:

8 2 3 ^ / 2 3 * + 5 1 * -

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

(A) 6, 1

(B) 5, 7

(C) 3, 2

(D) 1, 5

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 4

')' == '(', '(' == ')'

#Q. The Prefix Expression for an infix expression (A*B+C)/(D-(E*F)) is

A. +*/ABC-D*EF

B. +/*ABC-D*EF

C. /*+ABC-D*EF

D. /+*ABC-D*EF

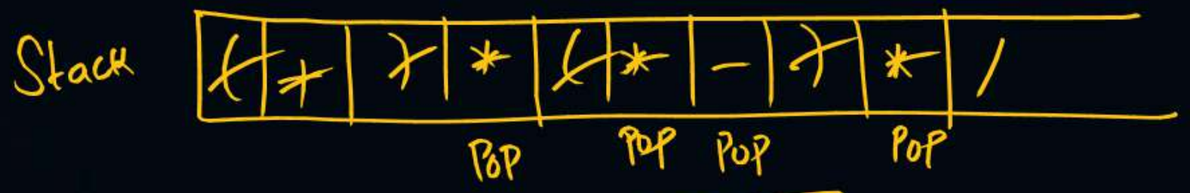
Intermediate Exp: FE*D-CBA*+/

Prefix exp: /+*ABC-D*EF

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 5

#Q. The postfix form of the expression is (A+B) * (C*D-E)*F/G is

A. AB+CD*E-FG/**
B. AB+CD*E-*F*G/
C. AB+CD*E-F**G/
D. AB+CDE*-*F*G/



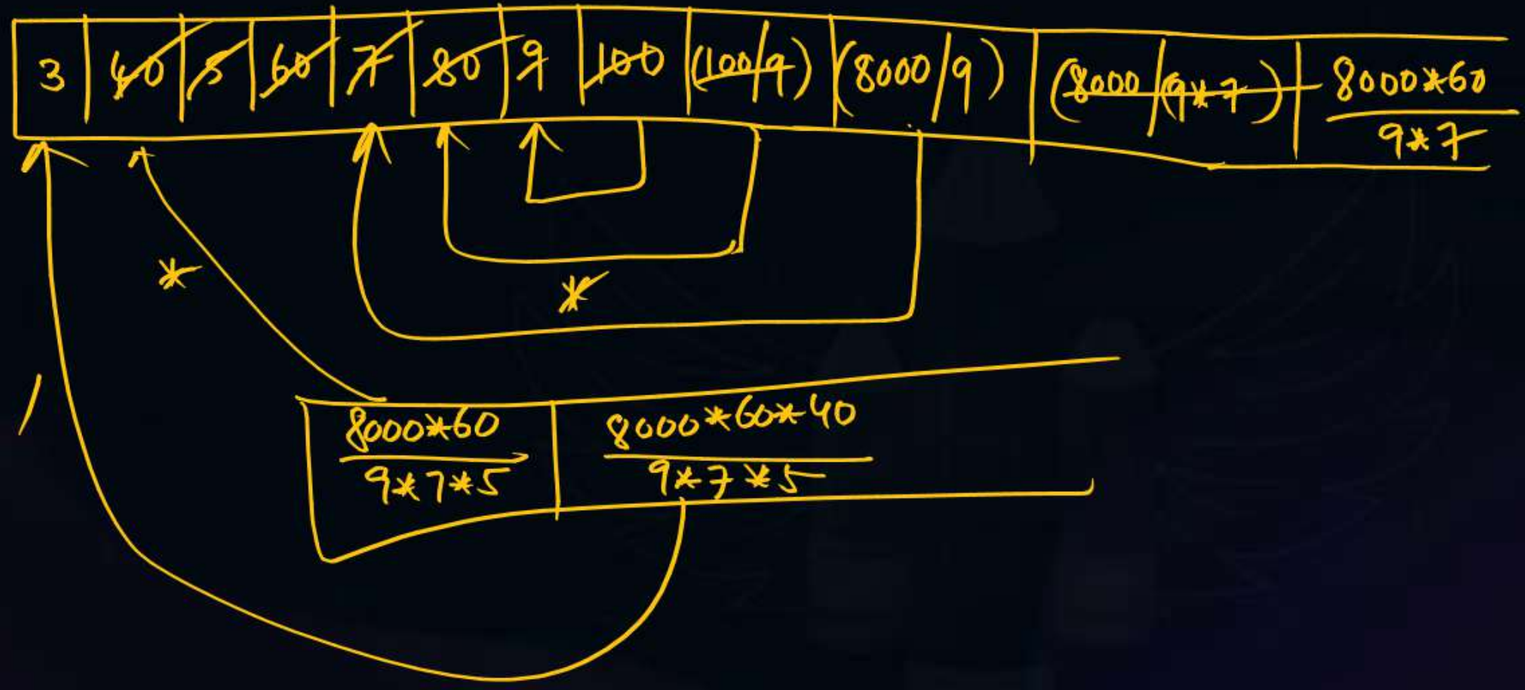
y: AB+cD*E−*F*G/

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 6

#Q. The Result after evaluating the Prefix Expression:

/ * / * / * / 100 9 80 7 60 5 40 3 is _____

A. 60000
B. 7500
C. 25000
D. 20000



$$\frac{8000 * 60 * 40}{9 \times 7 \times 5 \times 3}$$

$$\frac{8000 \times 160}{63} = 20317.$$

SUPER 1500+ - DSP- CLASS – 2 - Homework Question - 7

#Q. Consider the array implementation of stack:

Let Stack Max_SIZE = 10

If the array index starts with 0, the maximum value of top which does not cause stack overflow is?

A) 8
B) 9
C) 10
D) 11

```
def Push(S, item):

    if top == Max_SIZE - 1:
        Print('overflow')
        return

    top = top + 1
    S[top] = item.
```

Overflow Condition is TRUE when top = MAX_SIZE - 1

= top = 10 - 1 ⇒ top = 9

⇒ Max value of top, without overflow will be 8.

#Q.

Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?

(A) $h(i) = i^2 \bmod 10$

(B) $h(i) = i^3 \bmod 10$

(C) $h(i) = (11 * i^2) \bmod 10$

(D) $h(i) = (12 * i) \bmod 10$

$h(i) = i \,\%\, 10$

$h(23) = 23 \,\%\, 10 = 3$

$h(47) = 47 \,\%\, 10 = 7$

$h(X) = ABC \,\%\, 10$

⇒ LSB digit as result

Least significant bits (0 to 9 only)

i (LSB)	i^2 (LSB)	i^3 (LSB)	$11 * i^2$ (LSB)	$(12 * i)$ (LSB)
0	0	0	0	0
1	1	1	1	2
2	4	8	4	4
3	9	7	9	6
4	6	4	6	8
5	5	5	5	0
6	6	6	6	2
7	9	3	9	4
8	4	2	4	6
9	1	9	1	8

Uniformly distributing to all Buckets.

Buckets 2, 3, 7, 8 Not mapped

9, 3, 7, 8 Buckets Not mapped

Odd Buckets are Not mapped

Bucket Numbers

#Q. Consider a hash table with 100 slots. Collisions are resolved using Open Addressing. What is the probability that the first 4 slots are unfilled after the first 4 insertions?

(A) $(96 \times 96 \times 96 \times 96)/100^4$
(B) $(96 \times 95 \times 94 \times 93)/100^4$
(C) $(97 \times 96 \times 95 \times 94)/100^4$
(D) $(97 \times 96 \times 95 \times 94)/(4! \times 100^4)$

1st key : $\frac{96}{100}$

2nd key : $\frac{95}{100}$

3rd key : $\frac{94}{100}$

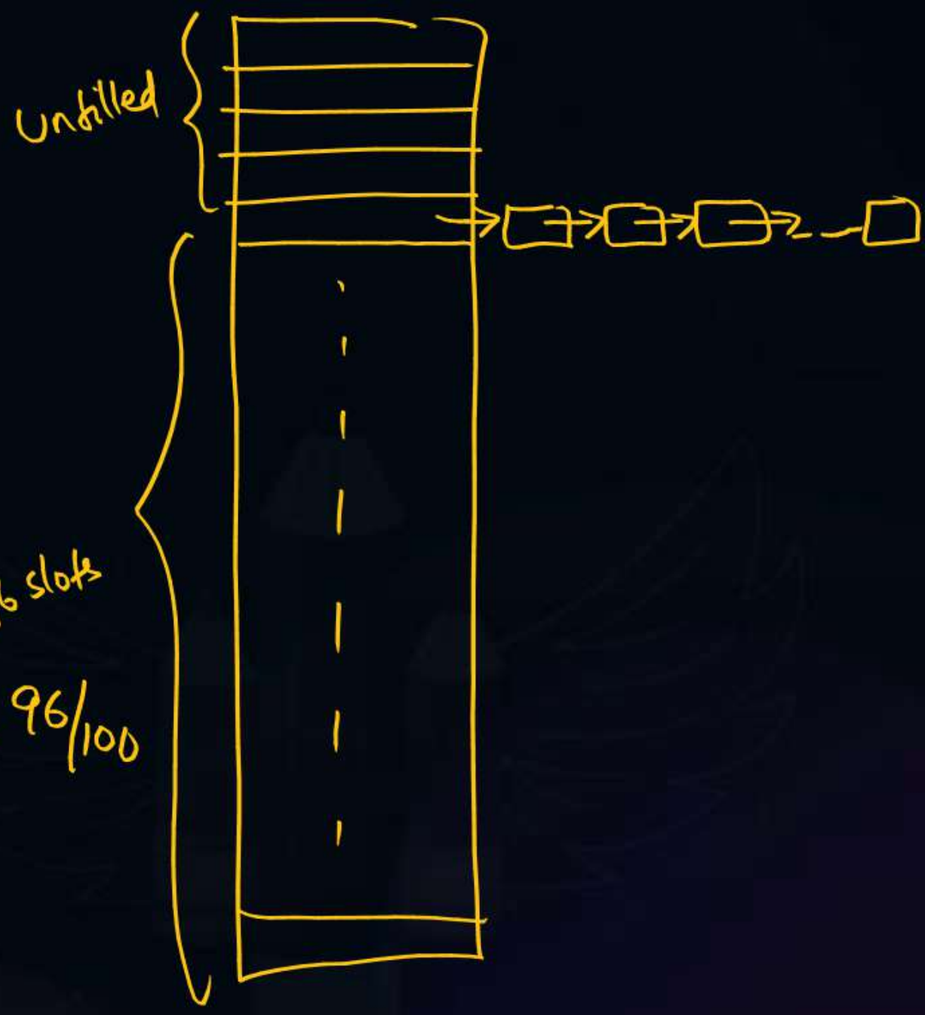
4th key = $\frac{93}{100}$

$\Rightarrow$ Total Probability = $\frac{96}{100} * \frac{95}{100} * \frac{94}{100} * \frac{93}{100}$

$= \frac{(96 * 95 * 94 * 93)}{100^4}$

#Q. Consider a hash table with 100 slots. Collisions are resolved using Separate Chaining Method. What is the probability that the first 4 slots are unfilled after the first 4 insertions?

(A) $(96 \times 96 \times 96 \times 96)/100^4$
(B) $(96 \times 95 \times 94 \times 93)/100^4$
(C) $(97 \times 96 \times 95 \times 94)/100^4$
(D) $(97 \times 96 \times 95 \times 94)/(4! \times 100^4)$



In chaining method, all keys also may be mapped to single slot.

So, 1st key = 96/100, 2nd key = 96/100, 3rd key = 96/100, 4th key = 96/100

$$\text{Total Probability} = \frac{96 \times 96 \times 96 \times 96}{100^4}$$

#Q. Consider a hash table with hash function H(i) = i mod 11 and following keys are hashed into the 24, 49, 20, 16, 23, 36, 34, 60 hash table, to handle the collision chaining is used, after inserting all the keys if new key is inserted then what is the probability that it hashed into empty slot _____. (Upto 2 decimal places)

$h(24) = 24 \% 11 = 2$

$h(49) = 49 \% 11 = 5$

$h(20) = 20 \% 11 = 9$

$h(16) = 5$

$h(23) = 1$

$h(36) = 3$

$h(34) = 1$

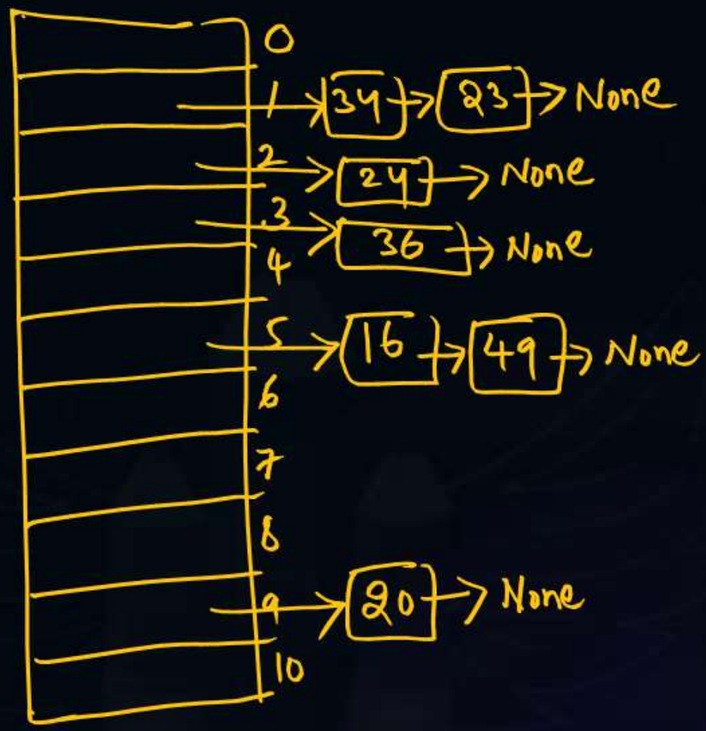
$h(60) = 5$

Number of Empty slots = 6

(0, 4, 6, 7, 8, 10)

$\text{Probability} = \frac{6}{11} = 0.545$

$\Rightarrow 0.54$



#Q. Suppose a circular queue of capacity (~~n - 1~~ n) elements is implemented with an array of n elements. Assume that the insertion and deletion operations are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are

(A) Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT

(B) Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR

(C) Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT

(D) Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

Full: front = (Rear+1) % size

Empty: front == -1 (GV)

front == Rear

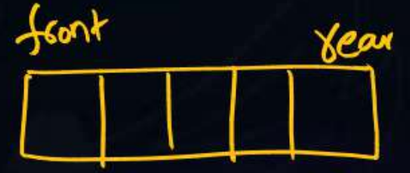
#Q. A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is CORRECT (n refers to the number of items in the queue)?

(A) Both operations can be performed in O(1) time

(B) At most one operation can be performed in O(1) time but the worst case time for the other operation will be Ω(n)

(C) The worst case time complexity for both operations will be Ω(n)

(D) Worst case time complexity for both operations will be Ω(log n)

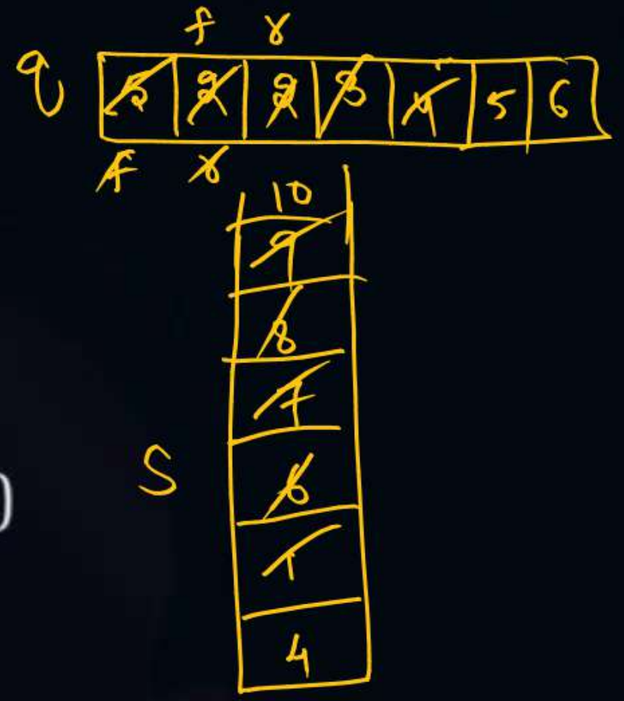


Enqueue( )
rear = rear + 1
Q[rear] = value
O(1)

Dequeue( )
item = Q[front]
front = front + 1
O(1)

#Q. Let q be a queue and S be a stack. The function dequeue and pop are the conventional operation that they return whatever they remove. Assume that q and S are initially empty and has been declared as an int.

```
enqueue (q, 5)
enqueue (q, 2)
push (S, 4)
push (S, 1)
for i in range(5):
        print (dequeue(q))
        print ( pop(S));
        enqueue (q, i + 2)
        push (S, i + 6);
```



i=0 enqueue(q, 2), Push(S, 6)

i=1 en(q, 3), Push(S, 7)

i=2 en(q, 4), Push(S, 8)

i=3 en(q, 5), Push(S, 9)

i=4 en(q, 6), Push(S, 10)

o/p: 5+1+2+6+2+7+3+8+4+9 = 47

What is the sum of all value printed by this code fragment _____.

SUPER 1500+ - DSP- CLASS – 3 - Homework Question - 1

#Q. Consider a hash table of size 10 Hashing is done using the hash function as :
Hf(Key) = key mod 8

Find the number of collisions ____ when the below keys are inserted in hash table in given order if linear probing is used.

25, 39, 46, 55, 89, 23, 68

SUPER 1500+ - DSP- CLASS – 3 - Homework Question - 2

#Q. Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.

(A) 8, _, _, _, _, _, 10
(B) 1, 8, 10, _, _, _, 3
(C) 1, _, _, _, _, _,3
(D) 1, 10, 8, _, _, _, 3

SUPER 1500+ - DSP- CLASS – 3 - Homework Question - 3

#Q. Consider a hash table of size 11 that uses open addressing with linear probing. Let h(k) = k mod 11 be the hash function used. A sequence of records with keys

43, 36, 92, 87, 11, 4, 71, 13, 14

is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. What is the index of the bin into which the last record is inserted?

(A) 2
(B) 4
(C) 6
(D) 7

SUPER 1500+ - DSP- CLASS – 3 - Homework Question - 4

#Q. Consider the circular queue given below which has FRONT =1 and REAR=5

Now perform the following sequence of operations on the queue

i. Enqueue F
ii. Dequeue 2 letters
iii. Enqueue G
iv. Enqueue H
v. Dequeue 4 letters
vi. Enqueue I

What would be the sum of the positions of the rear and front references?

2 mins Summary

- Hash Tables

- Queues

NEXT CLASS TOPIC: LINKED LISTS

THANK - YOU