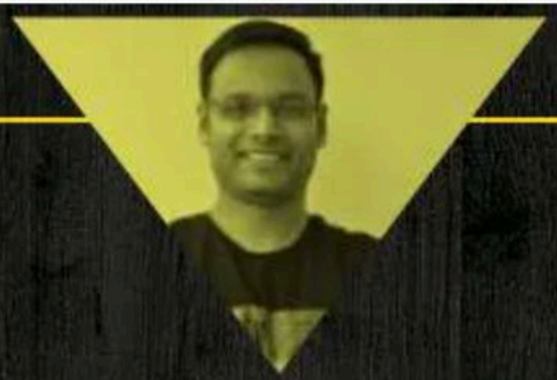
Doubt Clearing Session

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 36 • Jan 11, 2023

Data Structure:
Doubts & Stack

By: Vishvadeep Gothi



Hello!

I am Vishvadeep Gothi

I am here because I love to teach

DPP

Question 1 GATE-1996

Consider the following statements:

i. First-in-first out types of computations are efficiently supported by STACKS.

ii. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.

iii. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.

iv. Last-in-first-out type of computations are efficiently supported by QUEUES.

A. (ii) and (iii) are true
B. (i) and (ii) are true
C. (iii) and (iv) are true
D. (ii) and (iv) are true

Question 2

Which of the following permutations can be obtained in the output (in the same order) using a stack assuming that the input is the sequence 1, 2, 3, 4, 5 in that order?

a) 1, 2, 4, 5, 3
b) 4, 5, 3, 1, 2
c) 3, 5, 4, 2, 1
d) 1, 2, 5, 3, 4

A, C

Question 3 GATE-2014

Suppose a stack implementation supports an instruction $REVERSE$, which reverses the order of elements on the stack, in addition to the $PUSH$ and POP instructions. Which one of the following statements is TRUE (*with respect to this modified stack*)?

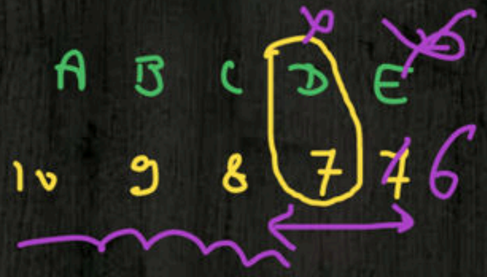
A. A queue cannot be implemented using this stack.
B. A queue can be implemented where $ENQUEUE$ takes a single instruction and $DEQUEUE$ takes a sequence of two instructions.
C. A queue can be implemented where $ENQUEUE$ takes a sequence of three instructions and $DEQUEUE$ takes a single instruction.
D. A queue can be implemented where both $ENQUEUE$ and $DEQUEUE$ take a single instruction each.

Question 4

How many valid and invalid stack permutations are there with a sequence of 6 inputs?

$$\text{valid} = \frac{{}^{2n}C_n}{n+1} = \frac{{}^{12}C_6}{7} = 132$$

$$\text{Invalid} = n! - \frac{{}^{2n}C_n}{n+1} = 6! - \frac{{}^{12}C_6}{7} = 588$$



Question GATE-1997

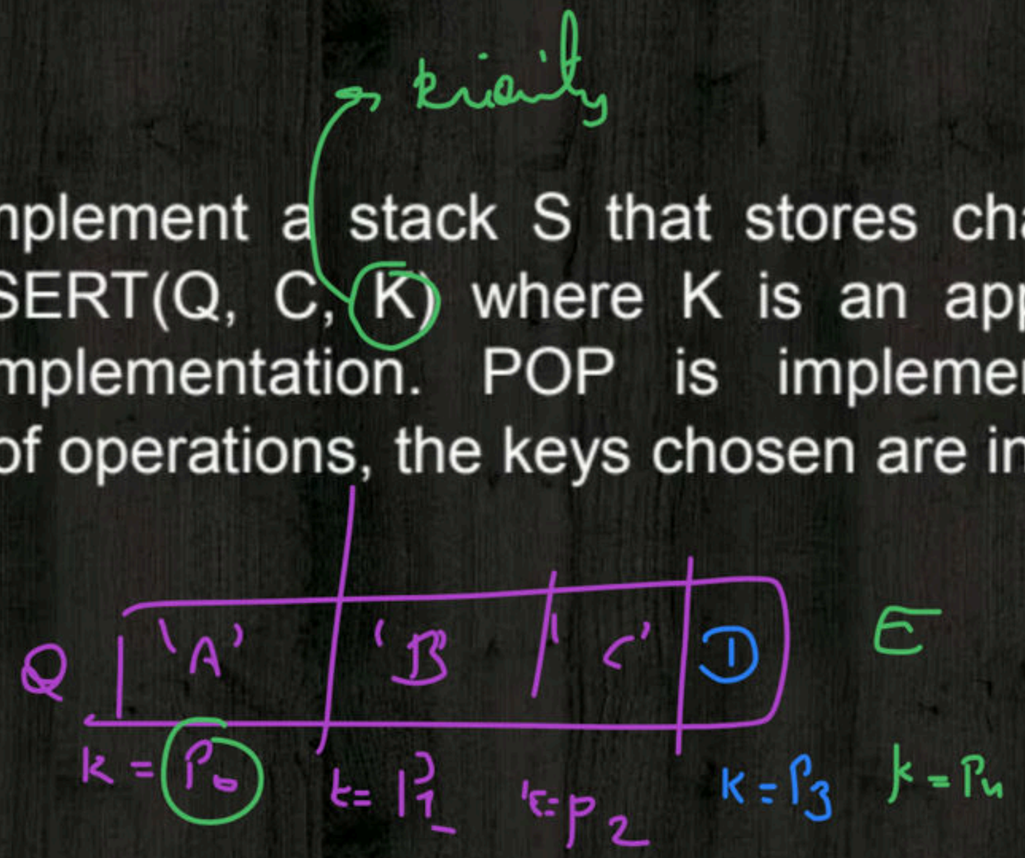
A priority queue Q is used to implement a stack S that stores characters. PUSH(C) is implemented as INSERT(Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN(Q). For a sequence of operations, the keys chosen are in

(A) Non-increasing order

(B) Non-decreasing order

(C) strictly increasing order

(D) strictly decreasing order



$\min(P_0, P_1, P_2)$

$P_3 = P_4$

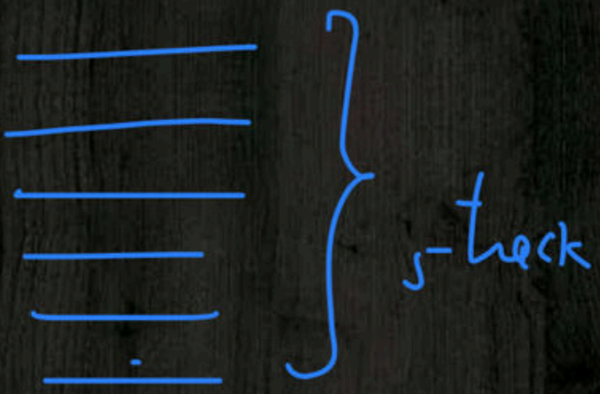
$P_0 > P_1 > P_2 > P_3$

DPP Question

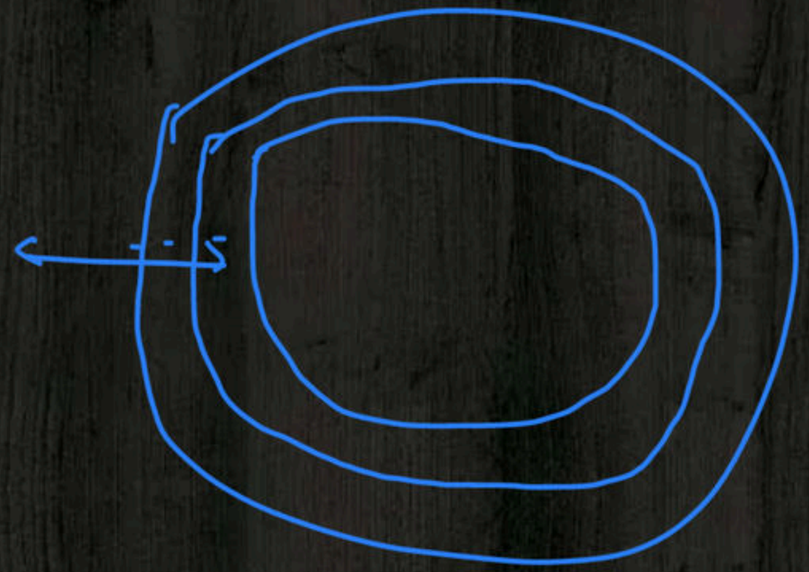
Ans = 7

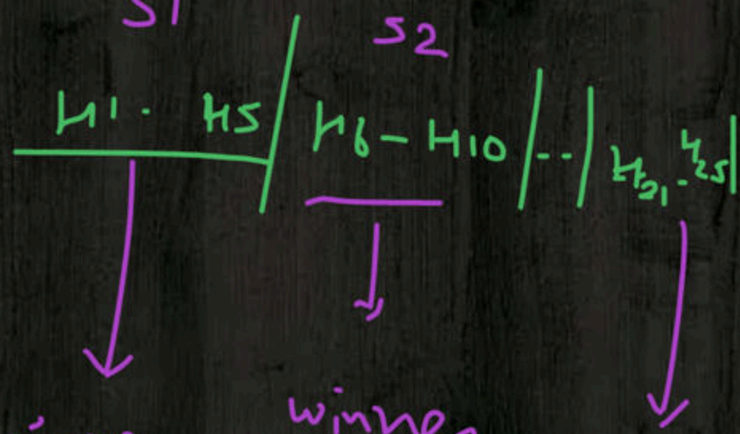
Horse and race question

4 horses ⇒ 25 horses



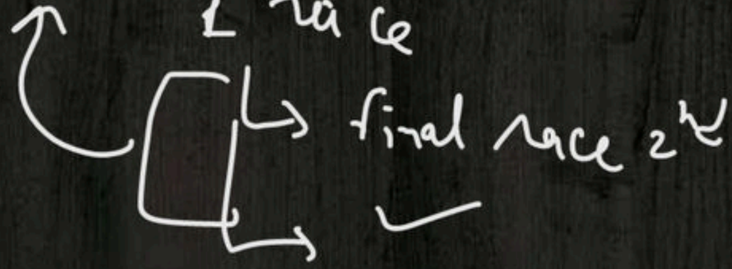
Best 2 horses (fastest)

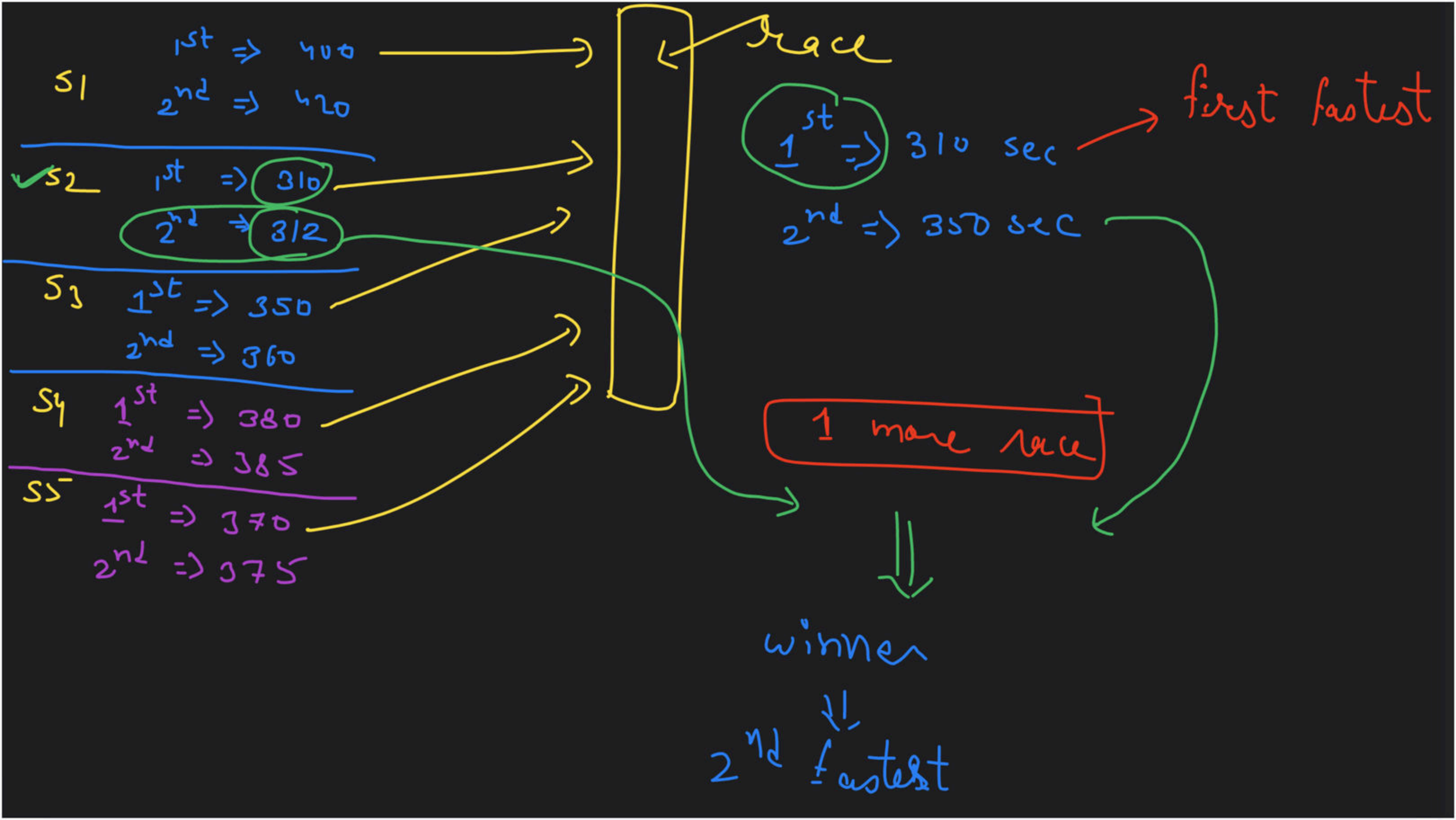




winner (fastest)

2nd fastest




race
S1
1st => 400
2nd => 420
S2
1st => 310
2nd => 312
S3
1st => 350
2nd => 360
S4
1st => 380
2nd => 385
S5
1st => 370
2nd => 375
1st
1 => 310 sec
first fastest
2nd => 350 sec
1 more race
winner
2nd fastest

Quiz Question

The minimum possible comparisons to find minimum and maximum both of an integer array of size 600 is _____?

$= 1.5n - 2 = 1.5 * 600 - 2$

$= 898$ Ans.

Quiz Question

Suppose you are playing game. You are expected to eliminate all n opponents from the game. The game started with n opponents against you. In the game when you eliminate 2 opponents then one new opponent is included in game. Elimination of 1 opponent and inclusion of 1 new opponent take constant time. The asymptotic complexity of elimination of all n opponent is?

Theta(1)

Theta(log n)

Theta(n)

CORRECT ANSWER

Theta(n^2)

operations

2+1 = 3

2+1 = 3

2+1 = 3

3

3

3

3

3

3

3

3

1

n = 10

1 2 3 4 5 6 7 8 9 10

7 8 9

3(n-1) + 1 = 3n - 3 + 1 = 3n - 2

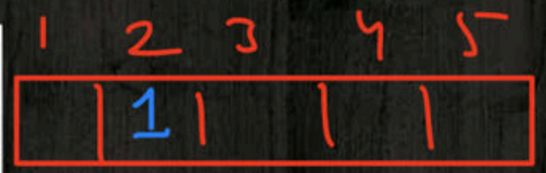
Quiz Question

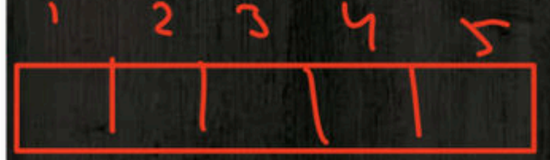
Consider an array A [1.....n]. It consists of a permutation of numbers 1 ..... n as elements. Now complete other arrays B[1 .....n] and C[1.....n] as follows:

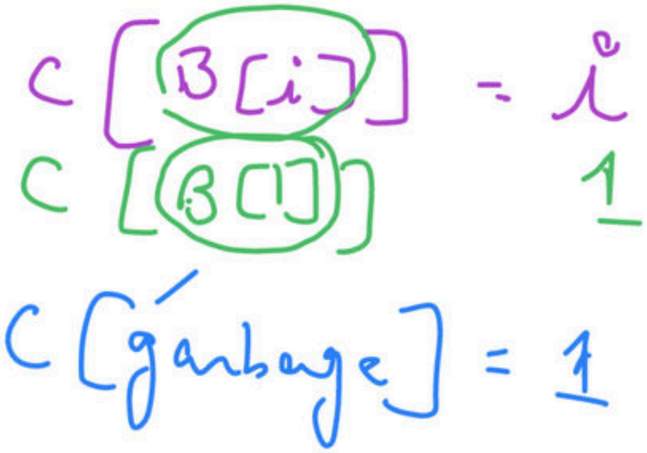
```
for (i = 1; i<=n; i ++)
{
B[A[i]] = i;
C[B[i]] = i;
}
```

Which of the following is true always?

- C will be sorted array in ascending order
- C is permutation of array A but not same as A
- C is identical as A
- None ✓







Quiz Question

Which of the following statement is/are not incorrect?

It takes linear time to traverse a linked list

CORRECT ANSWER

An element can be searched in logarithmic time in a sorted array

CORRECT ANSWER

It takes maximum n-1 number of comparisons to find minimum of an integer linked list of n elements

CORRECT ANSWER

A linked-list can be implemented using arrays

CORRECT ANSWER

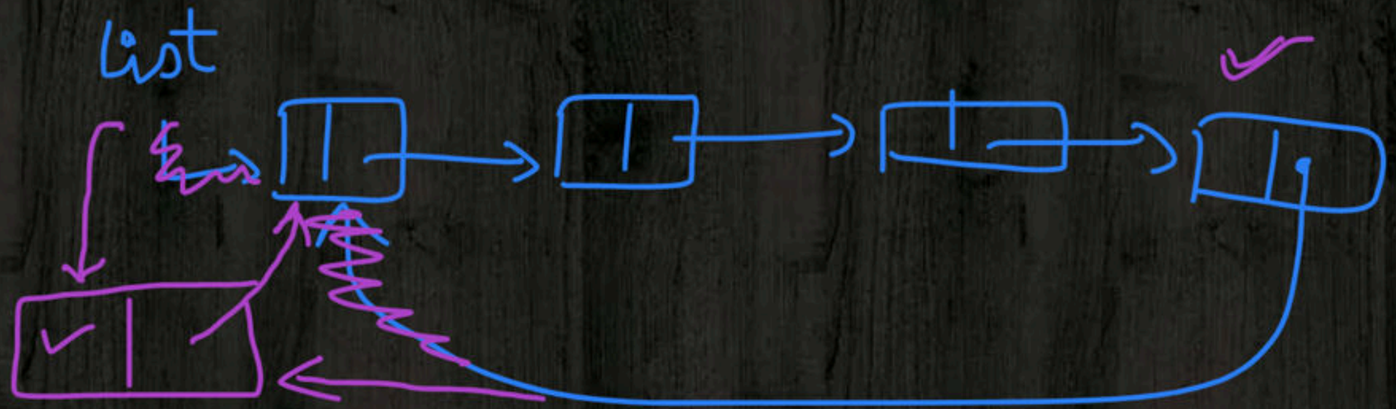
Quiz Question

Consider a 2-D array A[2:17][-14:2], which is stored in memory in column major order. The base address of array is 3000, each element takes x locations in the memory and the address of an element A[15][-2] is 3820. What is the value of x?

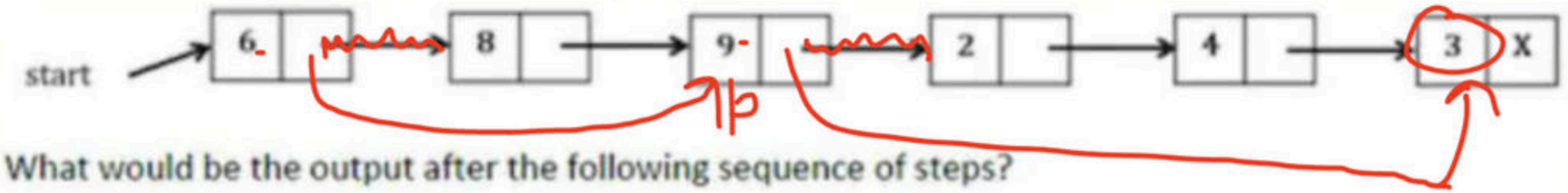
Quiz Question

Consider a circular singly linked list L which has only one list pointer, pointing to the first node the list. What is the run time complexity of inserting a new node at beginning of this list L?

Theta(1)	Theta(log n)
Theta(n)	Theta(n^2)



Quiz Question



What would be the output after the following sequence of steps?

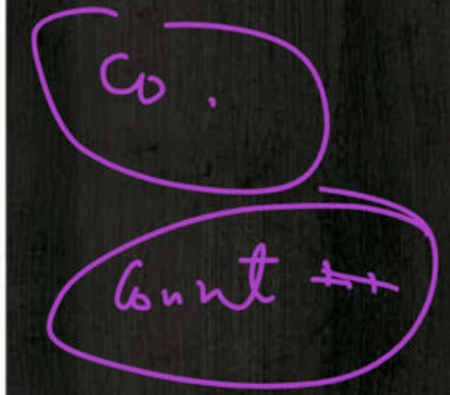
```
struct node{
int data;
struct node * link; };

void main(){

struct node *p;

p =start → link → link;

p → link  = p → link→ link→ link;

start → link = p;

printf("%c", start → link → link → data);
}
```

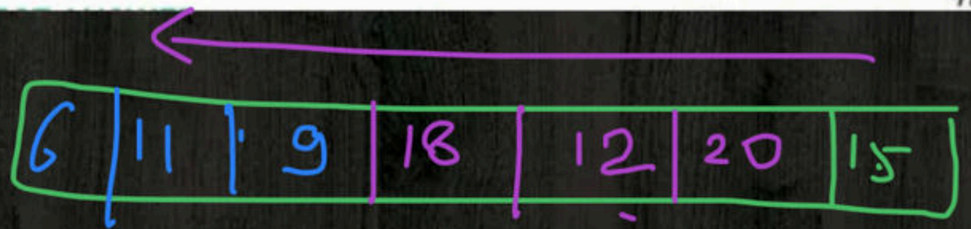
3

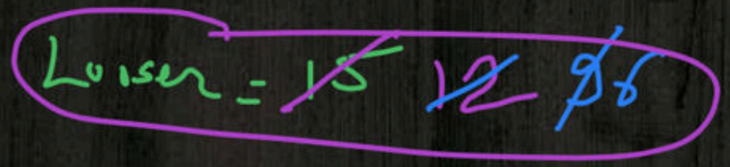
Quiz Question

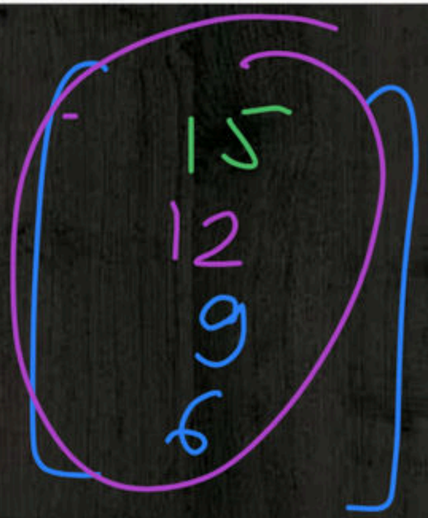
Consider an array A of size n(n>0), which contains integers in arbitrary order. Looser is an elements in the array which is lesser than all the elements from it's right in the array. What is the time complexity of best possible algorithm to find all the Looser winners in the array A?

- Theta(1)
- Theta(log n)
- Theta(n) ✓
- Theta(n^2)









Question GATE-2001

What is the minimum number of stacks of size n required to implement a queue of size n ?

(A) One

(B) Two

(C) Three

(D) n

Implementation of Queue Using Stack

Implementation of Queue Using Stack

Question GATE-2006

An implementation of a queue Q, using two stacks $S1$ and $S2$, is given below:

```
void insert (Q, x) {
    push (S1, x);
}
void delete (Q) {
    if (stack-empty(S2)) then
        if (stack-empty(S1)) then {
            print("Q is empty");
            return;
        }
        else while (!(stack-empty(S1))){
            x=pop(S1);
            push(S2,x);
        }
    x=pop(S2);
}
```

let n *insert* and $m(\leq n)$ *delete* operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of *push* and *pop* operations performed respectively in the process. Which one of the following is true for all m and n?

A. $n+m \leq x < 2n$ and $2m \leq y \leq n+m$

B. $n+m \leq x < 2n$ and $2m \leq y \leq 2n$

C. $2m \leq x < 2n$ and $2m \leq y \leq n+m$

D. $2m \leq x < 2n$ and $2m \leq y \leq 2n$

Multiple Stacks in Single Array

Multiple Stacks in Single Array

Question GATE-2004

A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (top1< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for “stack full” is

(A) (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2 + 1)

(B) top1 + top2 = MAXSIZE

(C) (top1= MAXSIZE/2) or (top2 = MAXSIZE)

(D) top1= top2 -1

Question

Consider an array of size n to implement m number of stacks (numbered 0 to m–1). The empty stack has following specifications:

for $0 \leq I < m$

$$\text{top}[i] = \text{bottom}[i] = i * \left\lfloor \frac{n}{m} \right\rfloor$$

for $i = m$

$$\text{bottom}[i] = n - 1$$

Where bottom is constant index and top is variable (moving) index.

Question continue

Fill in the blank:

1.

```
void push (int i, char item)
{
   if(_____)
            {
               printf("Overflow");
            }
   else
            {
               stack[++top [i]] = item;
            }
}
```

Question continue

Fill in the blank:

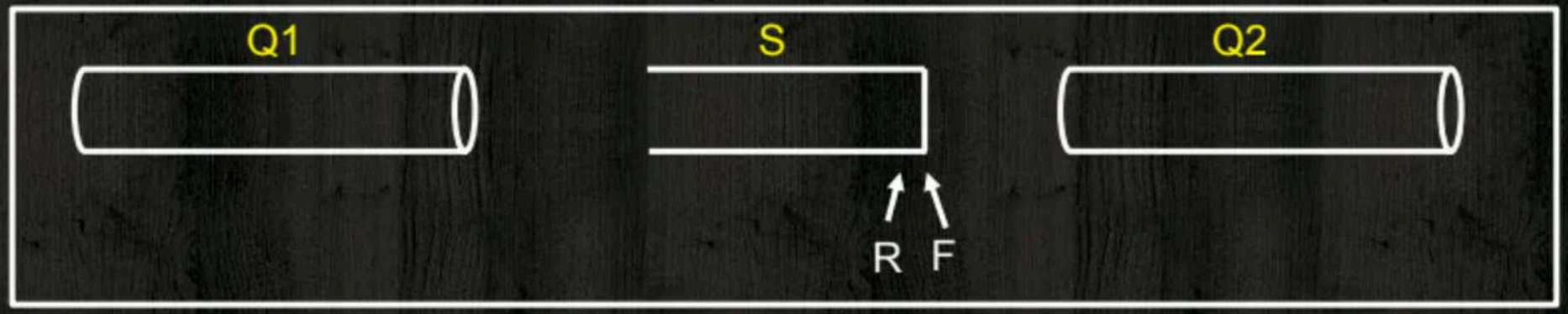
2.

```
void pop(int i)
{
   if(_____)
            {
               printf("Underflow");
            }
   else
            {
               stack[top[i]--];
            }
}
```

DPP

Question 1

Two queues Q1, Q2 and a stack S are implemented using a single array of size 20. Array has indexes 0 to 19 and has a base address $(504)_{10}$. The stack S and Queue Q2 are initially empty. The Q1, Q2 and S can be visualized as follows:



Size of Q1 = 10, Q2 = 6

The queue Q contains values: 5,9,3,4,0,7,9,4,0,6 from front to rear in that order. The array is of integers where each element takes 2 locations in memory. Consider the following program fragment executed on this array (on Q1, Q2 and S).

Question 1 cont..

```
Void doSomething()
{
 int x, y, z;
 while(!IsEmptyQueue(Q1))
 {
   x = dequeue(Q1);
   if(x ! = 0)
       PUSH(S, x);
   else
   {
      z=0;
      while(! IsEmptyStack(S))
              {
                 y=POP(S);
                 z+=y;
              }
   } } }
```

What are the addresses of elements 21 and 6 after execution of the function?

Question 2

Let **S** be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operations take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack-life of m as the time elapsed from the end of $Push(m)$ to the start of the pop operation that removes m from **S**. The average stack-life of an element of this stack is

A. $n(X+Y)$ B. $3Y+2X$ C. $n(X+Y)-X$ D. $Y+2X$

Question 3

Let Q denote a queue containing sixteen numbers and S be an empty stack. $Head(Q)$ returns the element at the head of the queue Q without removing it from Q. Similarly $Top(S)$ returns the element at the top of S without removing it from S. Consider the algorithm given below.

```
while Q is not Empty do
  if S is Empty OR Top(S) ≤ Head (Q) then
     x:= Dequeue (Q);
     Push (S, x);
  else
     x:= Pop(S);
     Enqueue (Q, x);
  end
 end
```

The maximum possible number of iterations of the **while** loop in the algorithm is _____.

Happy Learning

Expressions

↳ recursion

Prefix

Postfix

Infix

unacademy

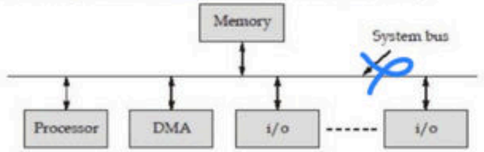
▲ 1 • Asked by Aira

Please help me with this doubt

Q. 4

? FAQ | ▶ Solution Video | Have any Doubt ?

Below diagram shows single bus detached DMA configuration for a system. How many times system bus used for a single data transfer using DMA (consider only data transfer, not command or status transfer).



A 1

B 2 — Your answer is Correct

Solution :
(b)
One access of system bus for data transfer from i/o to DMA and other for data transfer from DMA to memory.

C 3

D 0