



Problem solving - Part II

Course on Data Structure

Computer Science And Information Technology



Lecture Number : 36

Data Structure

By- Pankaj sir





#Q. Given a hash table T with 45 slots that stores 405 elements which occupied 32 slots.

Let p = load factor of T

Q = Space Utilization of T.

Then $(10p - 100Q) = \underline{\hspace{2cm}}$.

$$P = \frac{n}{m} = \frac{405}{45} = 9$$

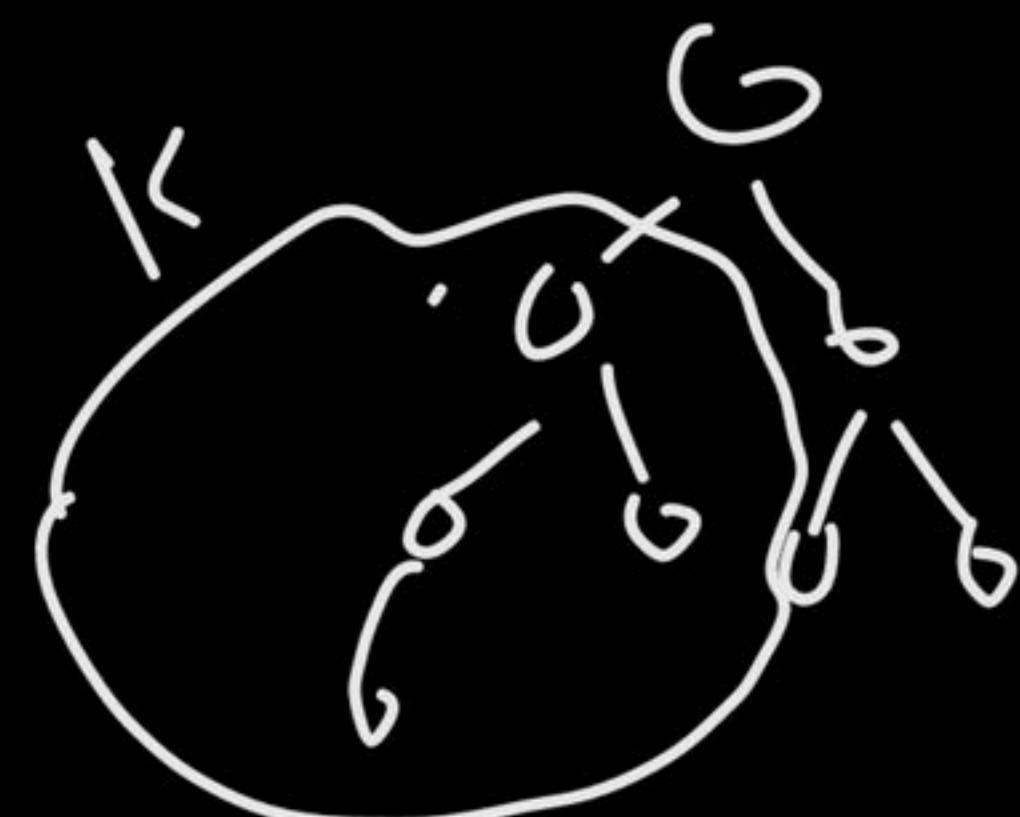
$$\begin{aligned} & 10 \times 9 - 100 \times \frac{32}{45} \\ & Q = \frac{32}{45} \\ & 18.85 \quad \checkmark \end{aligned}$$



#Q. For the elements 5 4 3 2 1 , how many max heap are possible ____.

No. of max-heap with n distinct keys:

$F(n)$: No. of max heap with n dist. keys



$$F(n) = \underbrace{1 \times \overset{n-1}{\underset{\text{LT}}{\underset{\uparrow}{\underbrace{C_k \times F(k)}}}}}_{\text{root}}$$

$$F(n-1-k)$$

$$F(n) = {}^{n-1}C_K \times F(K) \times F(n-1-K)$$

$$F(1) = 1$$

$$F(3) = 2$$

$$F(2) = 1$$

Q How many max heap can be created using
7 distinct keys.

$$F(7) = 1 \times \underbrace{^6C_3}_{\text{root}} \times F(3) \times F(3)$$

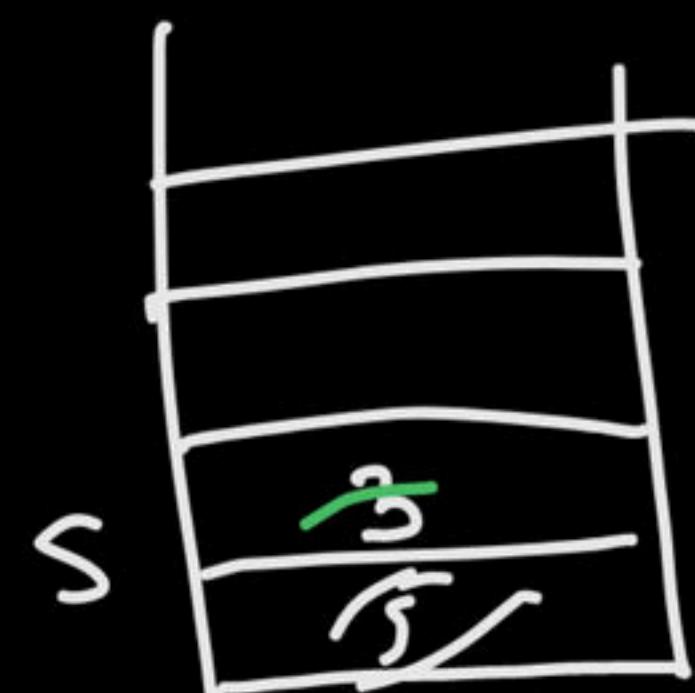
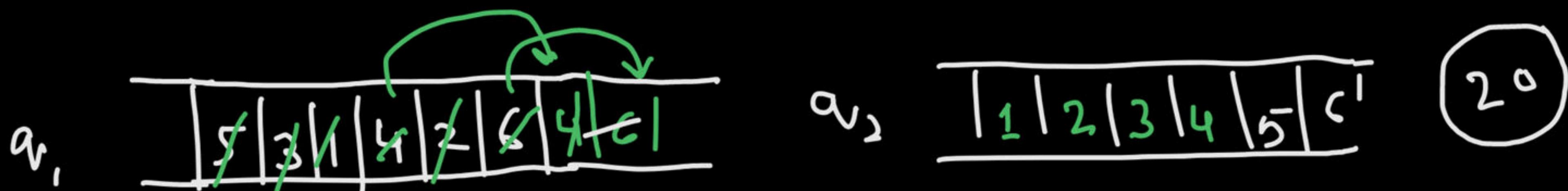
$$20 \times 2 \times 2 = 80$$





#Q. Consider 2 queues q_1, q_2 and a stack S operation allowed are enqueue, dequeue on q_1 & q_2 push & pop on S. q_1 has elements 5,3,1,4,2,6. 5 is at front. q_1 & S are empty.

Goal is to fill q_2 with 1, 2, 3, 4, 5, 6 from q_1 you can use stack min number of operation required are .



- 5 \rightarrow $deq_1 \rightarrow$ Push
- 3 \rightarrow $deq_1 \rightarrow$ Push
- 1 \rightarrow $deq_1 \rightarrow$ Enq_S, q_2
- 4 \rightarrow deq_1, Enq_S, q_1
- 2 \rightarrow deq_1, Enq_S, q_2

- Pop, Enq_S, q_2
- (C) 1 deq_1, Enq_S, q_1
- 4 deq_1, Enq_S, q_2
- 5 Pop, Enq_S, q_2

$\omega_1, deq_1, Enq_S, q_1$



#Q. Push operation takes 2 units of times, pop takes 1 unit of time, printing an element takes 3 units of time. Suppose S_1 , S_2 are 2 stacks where S_1 has elements $\{1, 2, 3, 4, 5\}$, 5 is at top and S_2 is empty. The min unit of time required to print the sequence 53142 is _____.
 (Note: only push, pop print can be used.)

Push-2

Pop-1

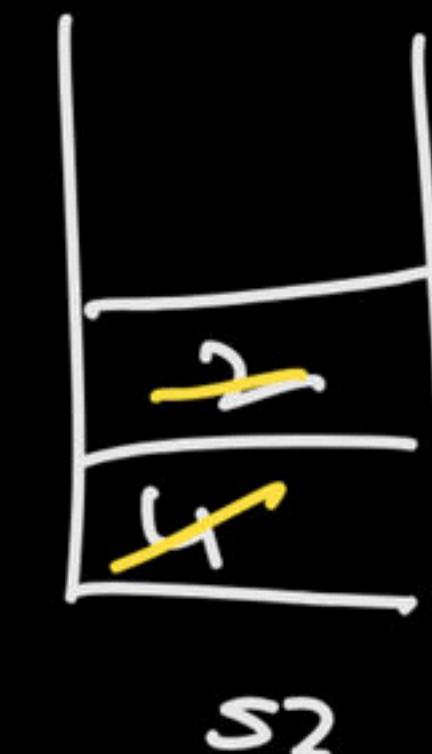
Printing-3

$$8 + 5 \times 3 + 3 \times 2$$

$$8 + 15 + 6$$

≥ 29

Pop(S_1) & Print



Pop(S_2) \Rightarrow Push(S_1)
 Pop(S_2) & Print

✓ 8 pop
 ✓ 5 3 1 4 2 5 print
 } bush

Pop(S_1) & Print
 - Pop(S_1) \rightarrow Push(S_2)

Pop(S_1) & Print

Pop(S_1) & Push(S_2)

Pop(S_1) & Print



#Q. $A = 9\ 5\ 2\ +\ *\ 15\ 3\ /-$

$B = -\ +/16^{\wedge}\ 23*\ 10\ 4\ A$

If above given expression evaluated using stack then, $B = \underline{\hspace{10cm}}$.

Already covered



#Q. Let

A = preorder successor of 17

B = preorder successor of 21

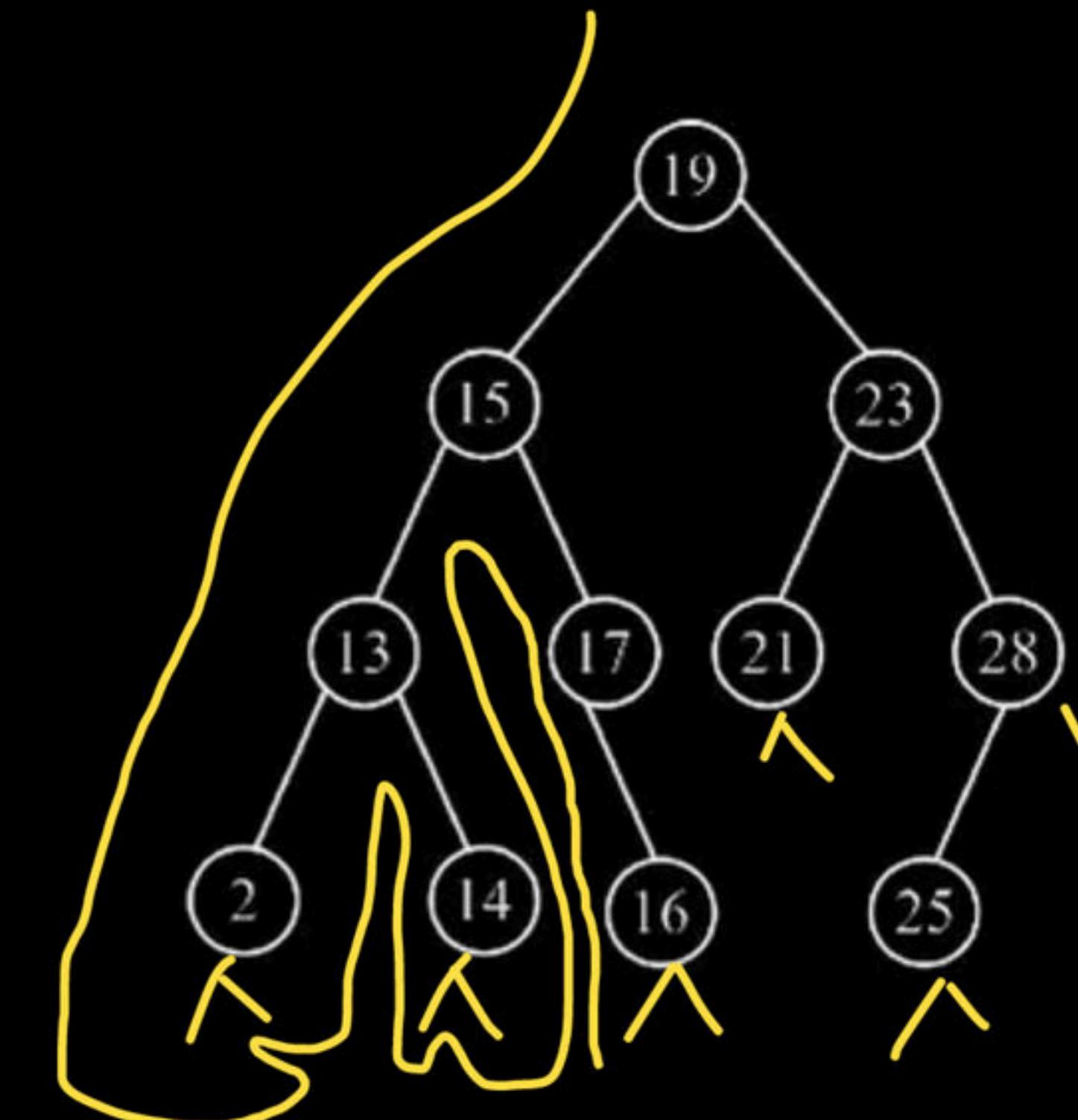
C = inorder successor of 19

Then $A/(B-C) = \underline{\hspace{2cm}}$.

$A \rightarrow 16$

$B \rightarrow 28$

$C \rightarrow 21$



Pre:

↙ ↘



#Q. Which of the following properties is/are correct for binary trees.

$$n_0 = n_2 + 1$$

- A Maximum number of nodes in a binary tree of height H is $2^{H+1} - 1$.
- B Total number of leaf nodes in a binary tree is equal to (total number of nodes with 2 children+1)
- C Maximum number of nodes at any level 'L' in a binary tree is 2^L .
- D Minimum number of nodes in a binary tree of height H is $H+1$.





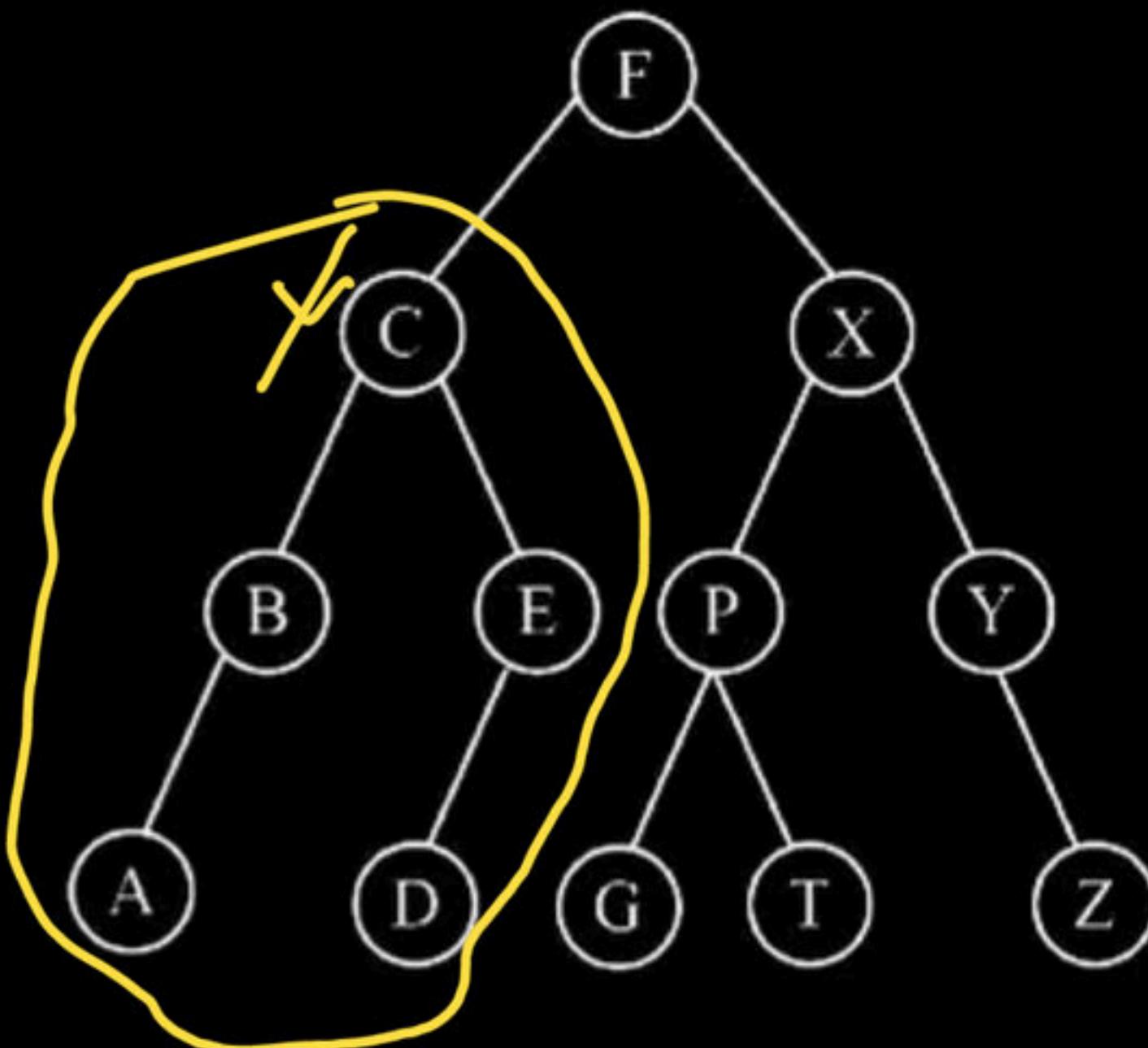
#Q. To apply binary search which of the following data structure is /are suitable.

- A Single linked list
- B Double linked list
- C array ✓
- D Circular linked list



#Q. Consider binary search tree,

If preorder traversal and post order traversal applied on left subtree of F and right sub tree of F respectively then which of the following option is correct?



A

ABCDE, GPTXYZ

E

ABDEC, XPGTYZ

C

CBAED, GTPZYX

D

ABCDE, XPGTYZ



#Q. The number of binary trees possible with the following given labelled nodes A,B,C,D,E is_____.

$$\frac{2^n T_n}{n+1} \times h!_0$$

$$\frac{16}{6} \times 5! \times 5!$$

$$\Rightarrow \underline{\underline{5040}}$$



#Q. Which of the following is/are correct corresponding to height balance tree?

A

Maximum height of AVL tree can not exceed $1.44 \log_2 n$ if there are 'n' nodes.

B

If height of AVL tree is 'h' maximum number of nodes in a tree can be $2^{\underline{h+1}} - 1$.

C

Minimum possible height of AVL tree is $\underline{[\log_2(n + 1)]}$ if there are 'n' nodes.

D

Minimum possible number of nodes in AVL tree of height 7 is 33.

h	0	1	2	3	4	5	6	7
$n(h)$	1	2	4	7	12	20	33	54



#Q. Consider the following three scenario of AVL tree,

P = Height of AVL tree is 9 then maximum no. of nodes are

Q = Minimum possible number of nodes in AVL tree of height 9 is

R = Minimum possible height of AVL tree with 16384 nodes.

$$\log_2 (P+7Q+2R-4) = \underline{\hspace{2cm}}$$

$$\frac{9+1}{2} = 1023$$

$$P \Rightarrow$$

$$Q \Rightarrow 143$$

$$R = 14$$

$$\left\lfloor \log_2 (16384+1) \right\rfloor$$

$$\Rightarrow 14$$

h	6	1	2	3	4	5	6	7	8	9
$n(h)$	1	2	4	7	12	20	33	54	88	143

$$P + 7Q + 2R - 4$$

$$\log_2 \left(1023 + 7 \times 143 + 2 \times 14 - 4 \right)$$

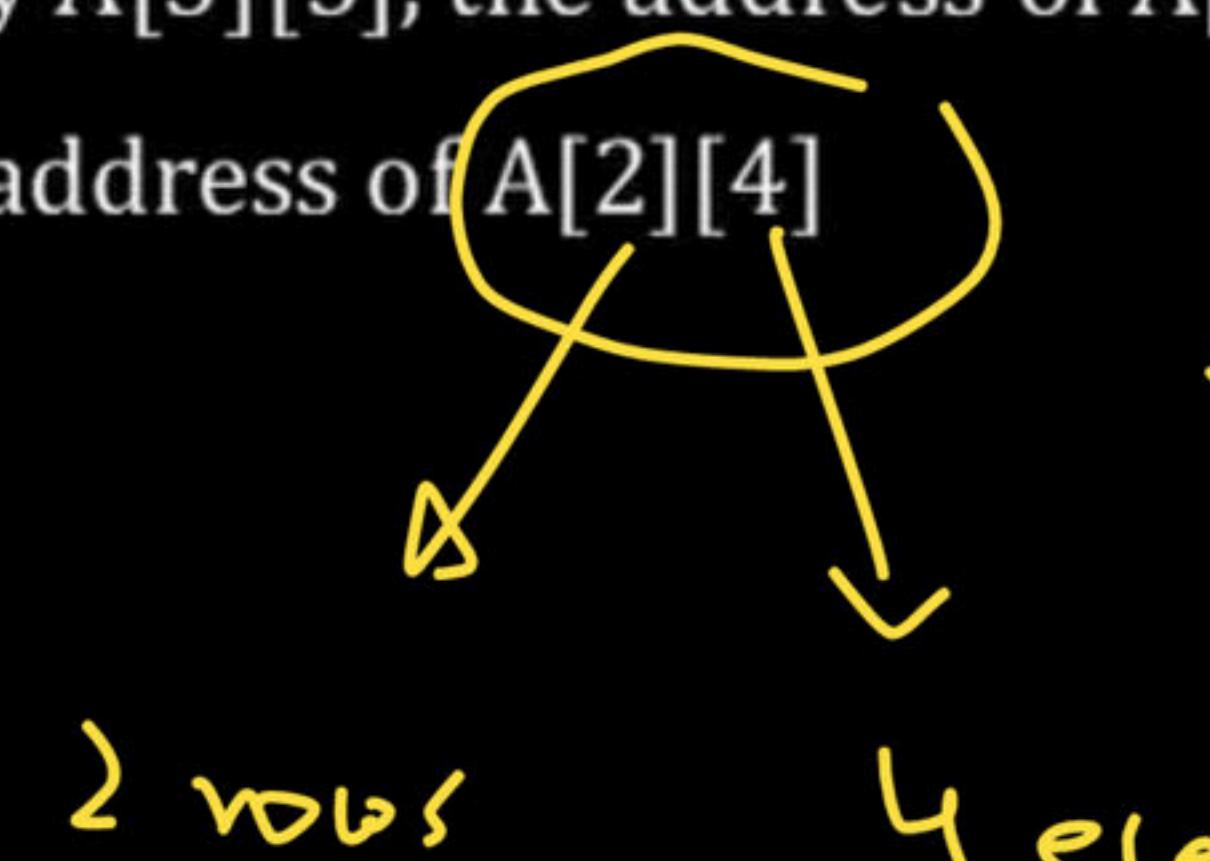
$$\log_2 2048$$

$$\log_2 2^{11} \Rightarrow 11 \cdot \log_2 2 \Rightarrow 11$$



#Q. Consider an array A[3][5], the address of A[0][0] is 200 & A[0][1] is 212
then what is the address of A[2][4]

- A 380
- B 368
- C 370
- D 382



$$\begin{aligned} \text{size of 1 ele} &= 212 - 200 \\ &= 12 \text{ bytes} \end{aligned}$$

$$(2 \times 5 + 4) \Rightarrow 14 \text{ elements}$$

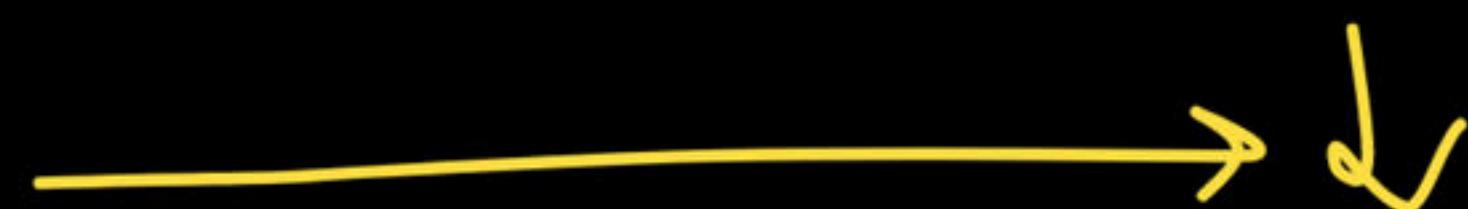
$$\text{Memory occupied} = 14 \times 12 = 168 \text{ bytes}$$

$$200 + 168$$



#Q. Consider a single linked list of n elements, then the min order of time to interchange m^{th} and l^{th} elements.

3^{rd} 10^{th}



- A max (m, l)
- B max (n, m, l)
- C min (m, l)
- D $l + \min(m, l)$



#Q. Array of 1023 elements used to construct the binary heap with starting index '0'. If the right child node is stored at index 510 then its parents parent node is at index ____.

right child $\Rightarrow 2i+2$



$$2i+2 = 510$$

$$2i = 508$$

$$\boxed{i = 254}$$



#Q. Which data structure is most efficient to find top k largest items out of n items stored in file?

- A Max heap ✓
- B Min heap
- C BST ✓
- D Sorted array ✓

Analysis

$n \log n$

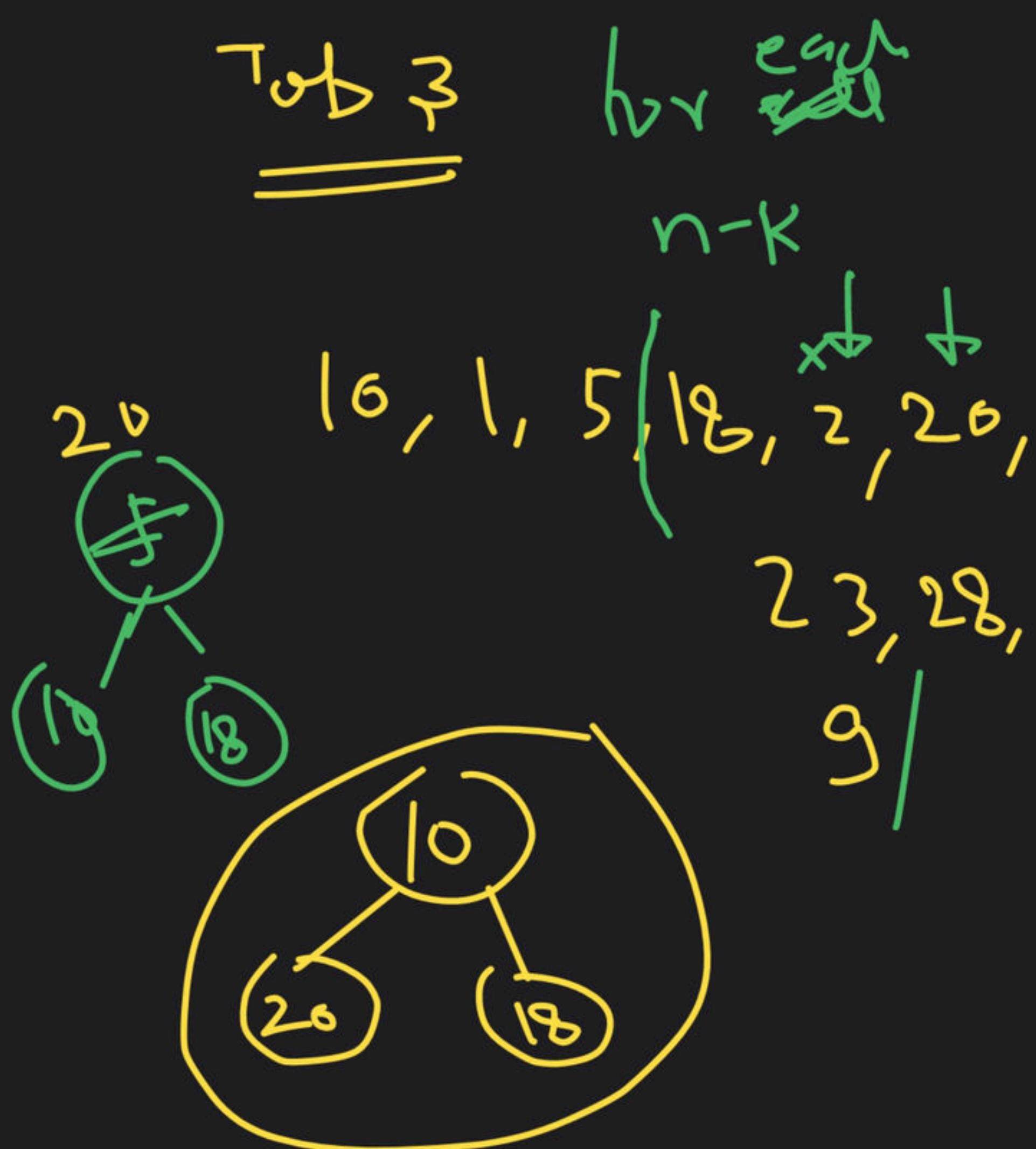
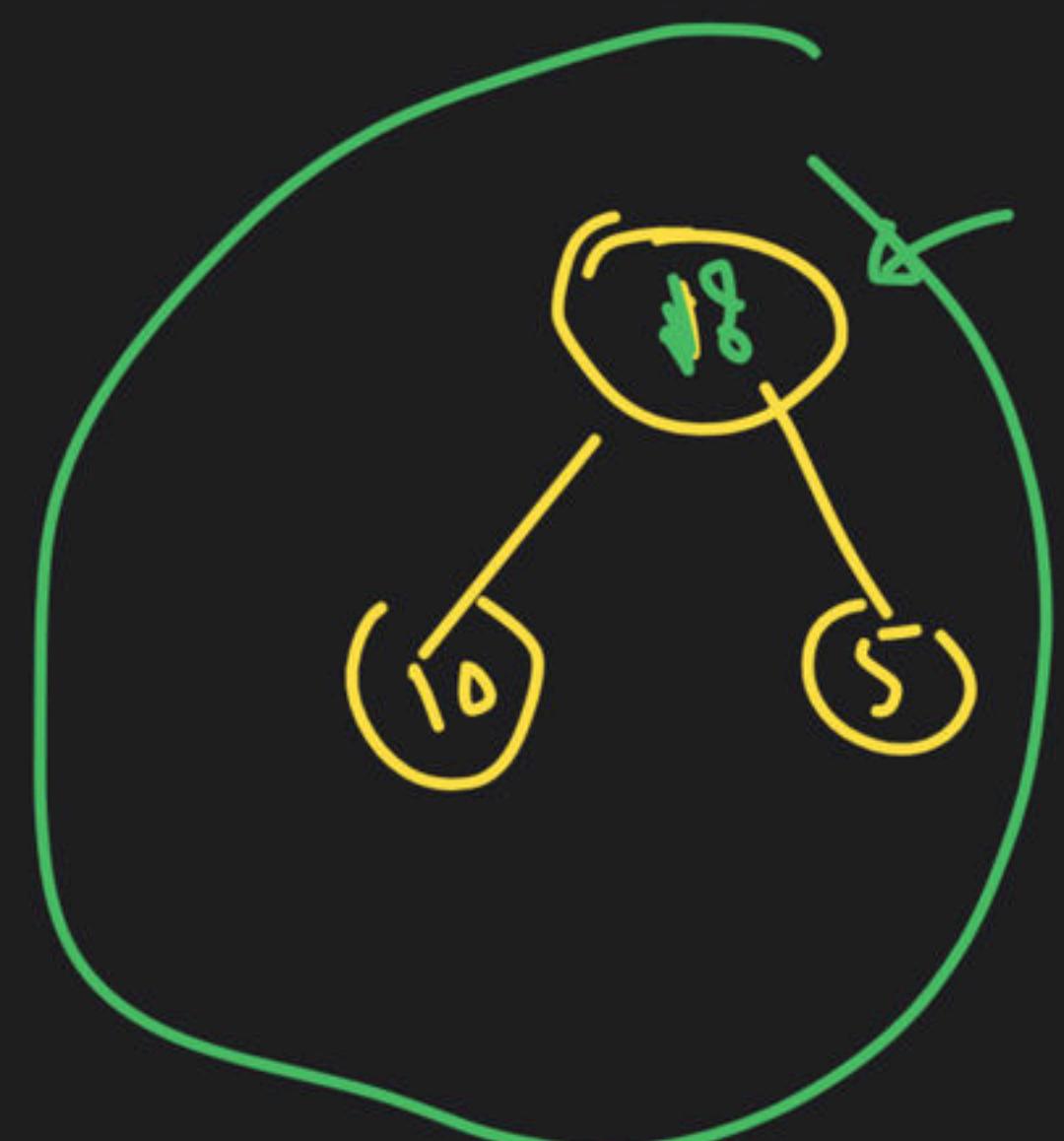
$n \log n + O(n)$

$O(n)$

$O(k \cdot \log n)$

$\Rightarrow O(k \cdot \log_2 n)$

$3 \Rightarrow$ size min heap



Heap $\Rightarrow O(k)$

$(n-k) \log_2 k$

$$O(k) + O((n-k) \cdot \log_2 k)$$

\uparrow

T

K smallest element

↗ min-heap

K largest element ↗ min-heap
→



#Q. The height of a binary tree is the maximum number of edges in any root node to leaf node path. The maximum number of nodes in a binary of height 15 is_____.

$$2^{15+1} - 1$$

$$= 2^{16} - 1$$

$$= 65536 - 1$$

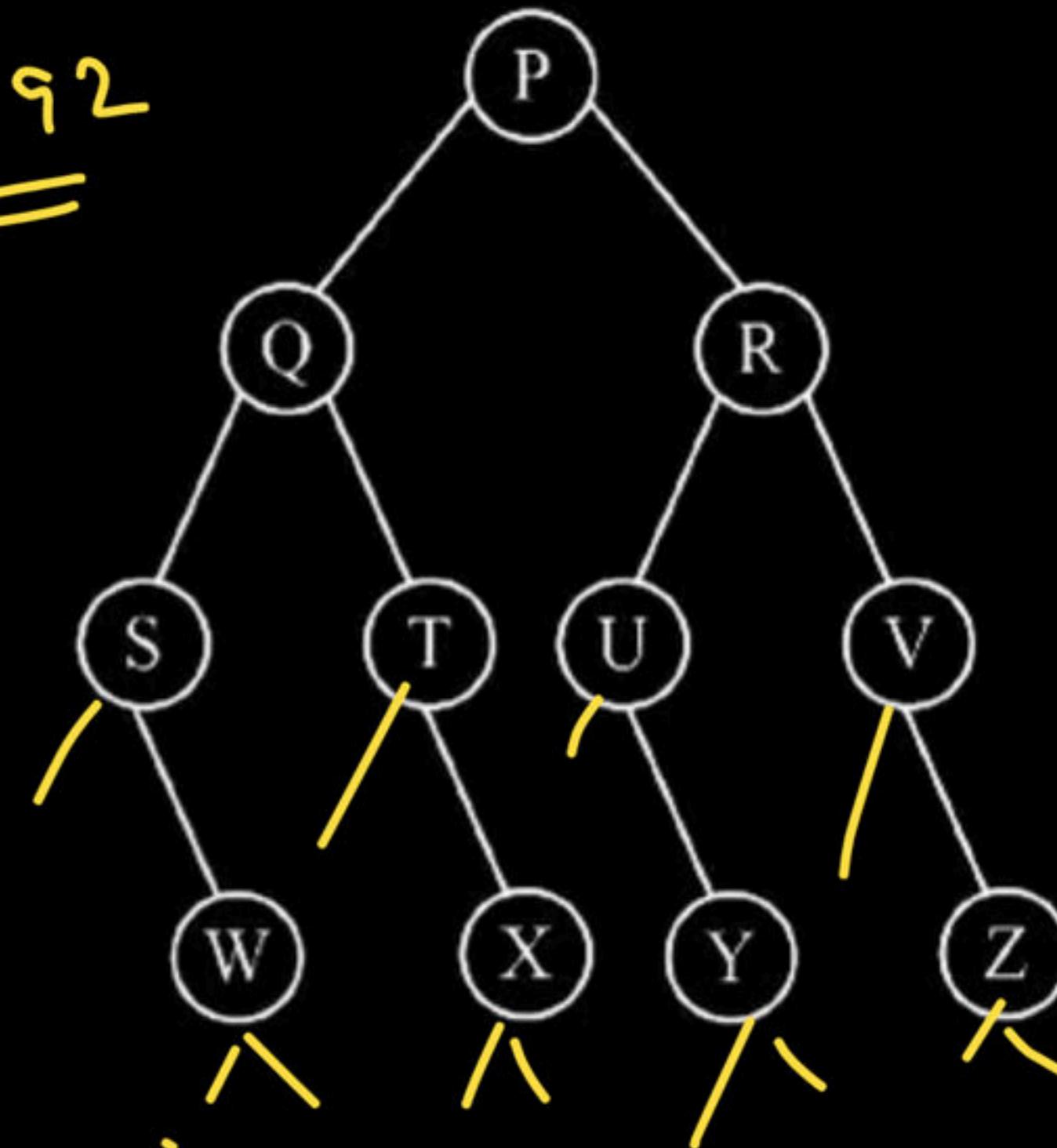
= 65535



- 21/14-17

#Q. Consider the following tree

$\rightarrow -18 \frac{92}{92}$



If the postorder traversal of above tree is,

$3 - 9 + * 14 + 17 - + /$. Then the SWQXPYVZ

= _____ (round off two decimal places)

WSXTPYVZVRP

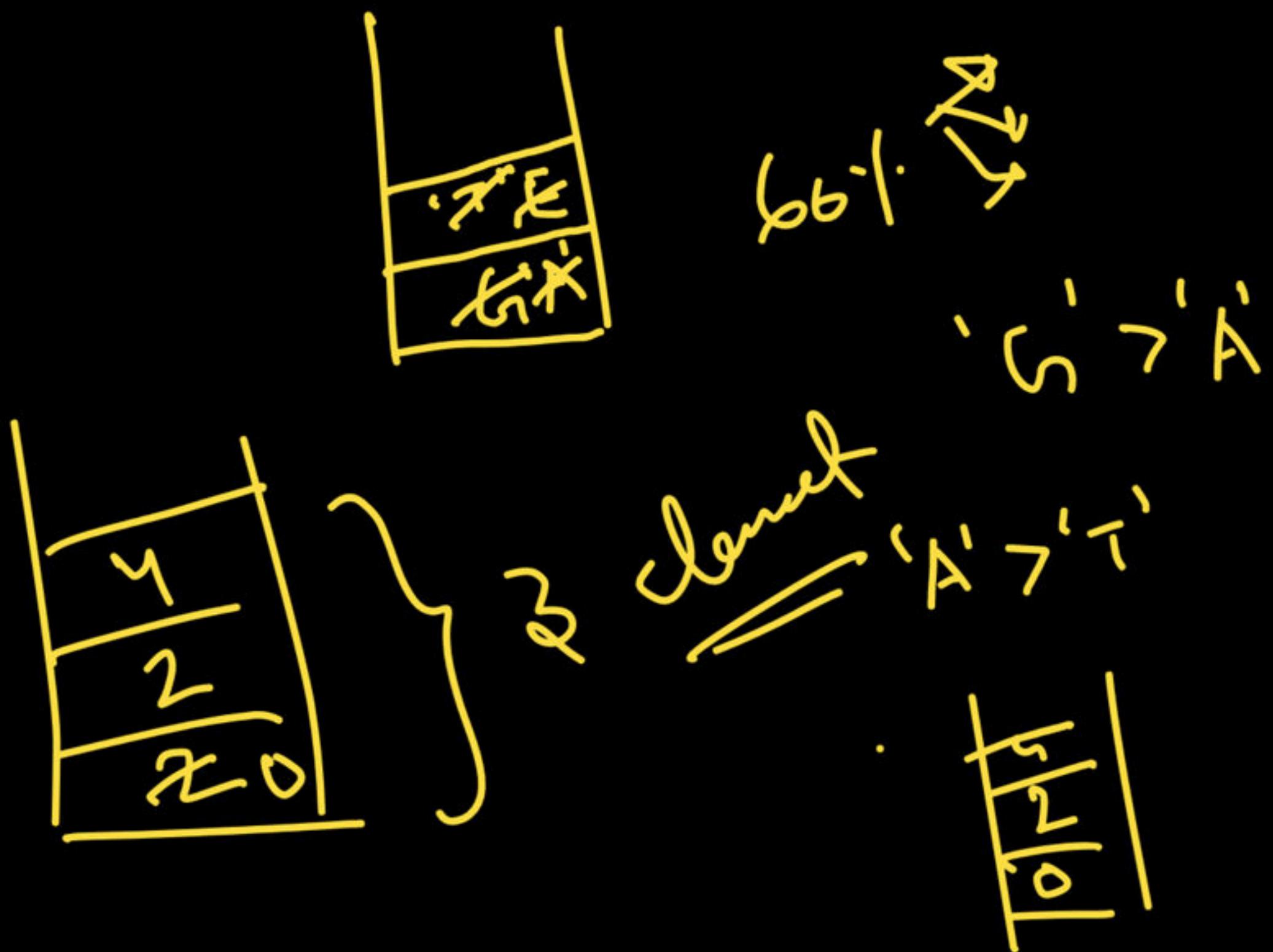
$3 - 9 + * 14 + 17 - + /$

$-3 * 9 / 14 - 17$



#Q. Assume P is an array of characters given as P ['G', 'A', 'T', 'E', '2', '0', '2', '4'] with indices going from 0 to n-1 (where n = 8). Consider running the following operation on stack 'S' with input array P given as above. The size of the stack 'S' at the end of the execution of below code is ____.

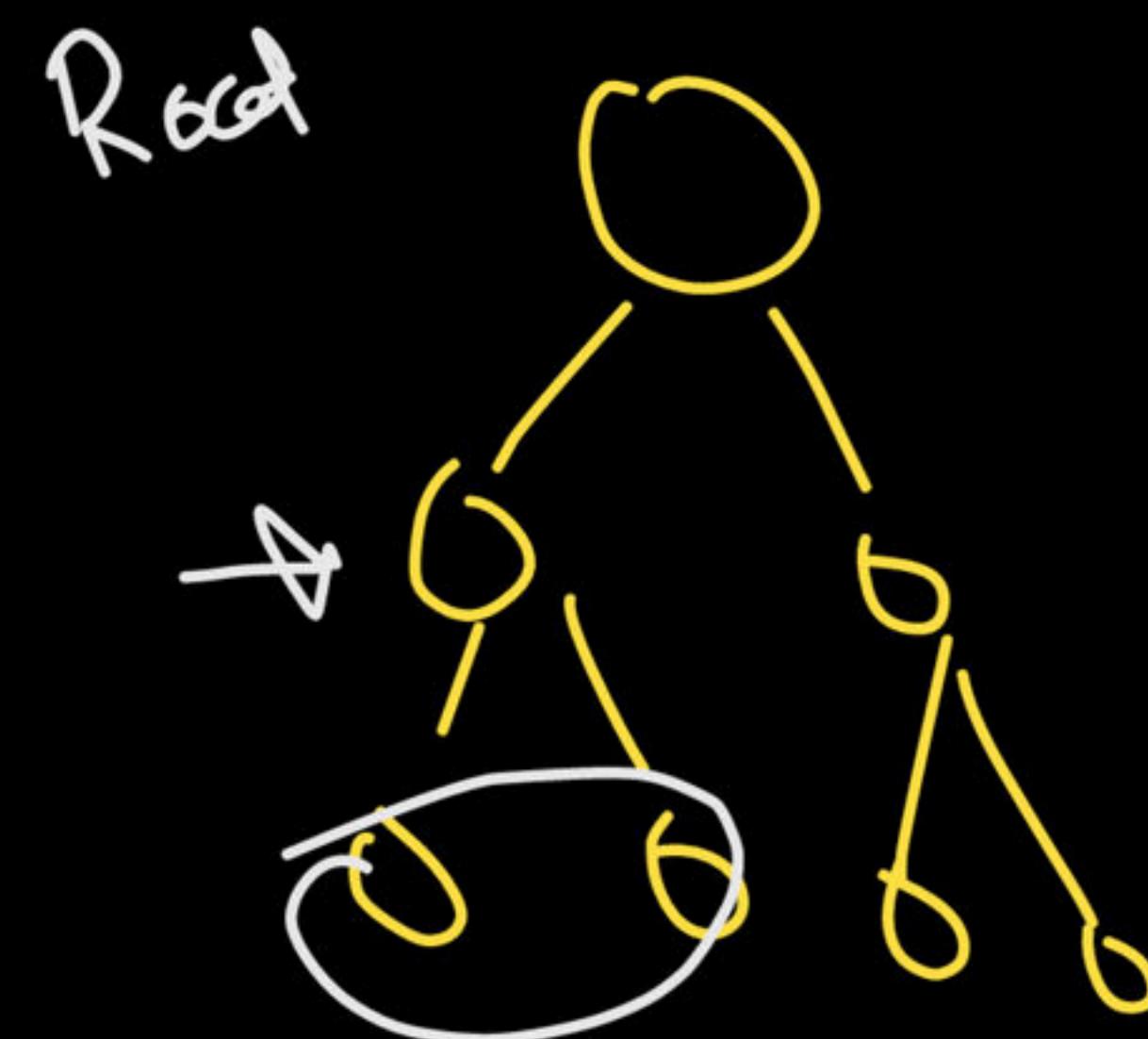
```
stack S = new stack();
int i = 0;
while (i < n) {
    if (S.is empty ()) {
        S. Push (P [i]);
        i++;
    }
    elseif (S. top > P[i]) {
        S. pop ();
    }
    else {
        S. Push (P[i]);
        i++;
    }
}
```





#Q. Which traversal is the most suitable for deleting all the nodes in a binary tree?

- A Inorder
- B Preorder
- C Postorder ✓
- D Any Traversal





#Q. A 2 dimensional array A[3][5] is stored in column major order and the address of A is 200 and array contains element from 1-15 stored in the matrix in increasing order in row major order. What is the address of the location element 10 is stored ? _____.

4. w
= =



#Q. Stack s1 has elements 5 4 3 2 1 in the sequence where 1 is at top. s2 is an empty stack. when an element popped from s1, it can be either printed or pushed into s2 not both. But when you pop from s2 it can only be printed. Which permutation is not possible.



1 2 3 4 5 ✓



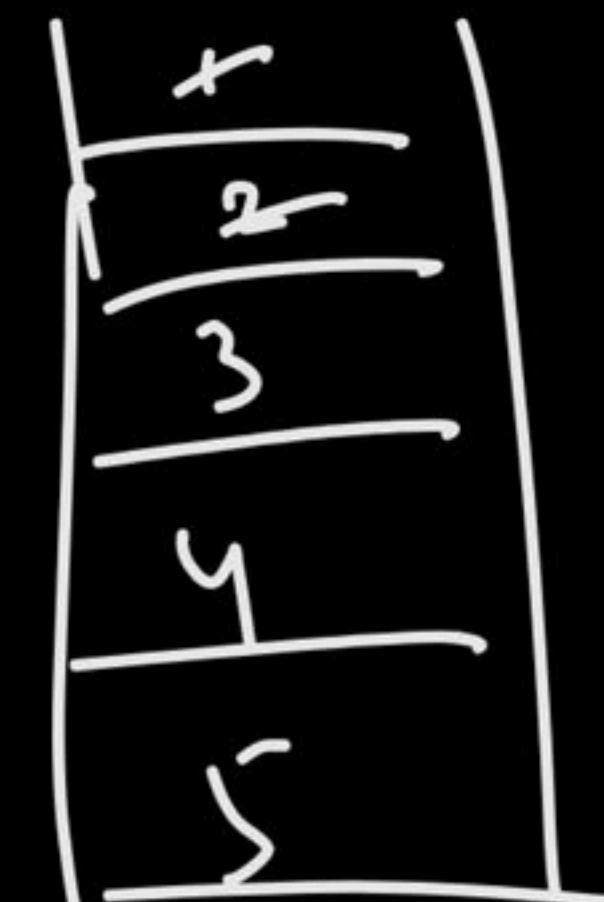
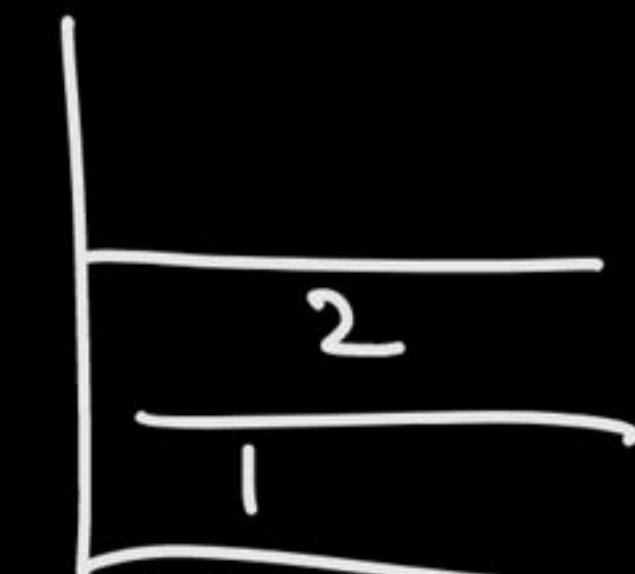
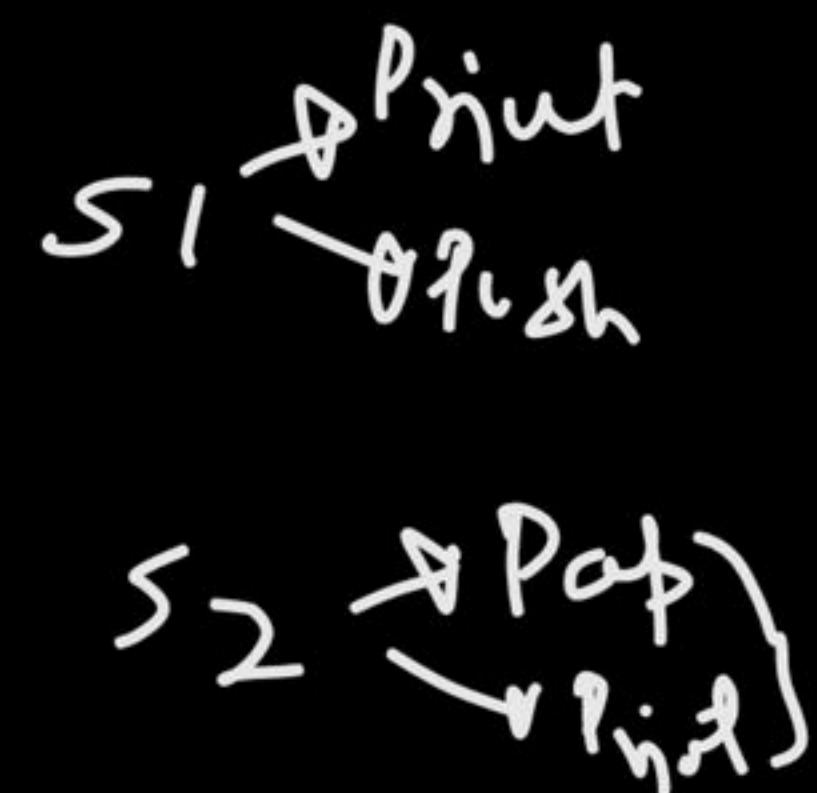
3 4 5 2 1 ✓



3 4 5 1 2 ✗



1 3 5 4 2 ✓

s₁s₂



Q. Consider the following C function, which is intended to return the index of the first negative number in an array of integers with length n. If no negative number is found, it should return -1:

```
int DoSomething(int[] array, int n) {  
    int i = 0;  
    while (P) {  
        i++;  
    }  
    if (i == n)  
        return -1;  
    return i;  
}
```

EASY

Which of the following conditions should replace P to ensure the function works as intended?

- A array[i] >= 0 && i < n
- B array[i] > 0 && i <= n
- C array[i] <= 0 || i < n
- D array[i] == 0 && i < n



Q. Consider the following code fragment where the heads of two sorted linked lists are passed as arguments:

```
struct node* fun(struct node* a, struct node* b) {  
    struct node* result = NULL;  
    if (a == NULL) return (b);  
    else if (b == NULL) return (a);  
    if (a->data < b->data) {  
        result = a;  
        result->next = fun(a->next, b);  
    }  
    else  
    {  
        result = b;  
        result->next = fun(a, b->next);  
    }  
    return result;  
}
```

- A Merges the two sorted linked lists into a single sorted list
- B Appends the second list to the first list without sorting
- C Returns the intersection of the two lists
- D Returns the union of the two lists without duplicates

2 days Linked list
= Code

45 question
+
46

Which of the following statements is true regarding the function fun()?



Q. The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers in the given order. What will be the contents of the list after the function completes execution?

```
struct node
{
    int value;
    struct node *next;
};

void rearrange(struct node *list)
{
    struct node *p, *q;
    int temp;
    if (!list || !list->next) return;
    p = list; q = list->next;
    while(q && q->next) {
        temp = p->value;
        p->value = q->next->value;
        q->next->value = temp;
        p = q->next->next;
        q = p ? p->next : 0;
    }
}
```

- A 3, 2, 1, 6, 5, 4, 7
- B 1, 3, 2, 5, 4, 7, 6
- C 2, 1, 4, 3, 6, 5, 7
- D 3, 4, 5, 6, 7, 1, 2

Initial List: 1, 2, 3, 4, 5, 6, 7

What will be the final contents of the list?



Q What will be the contents of the list after the function completes execution?

```
struct node {  
    int value;  
    struct node *next;  
};  
void rearrange(struct node *list)  
{  
    if (!list || !list->next || !list->next->next) return;  
    struct node *p1 = list, *p2 = list->next, *p3 = list->next->next;  
    while (p1 && p2 && p3)  
    {  
        int temp = p1->value;  
        p1->value = p3->value;  
        p3->value = temp;  
        p1 = p3->next;  
        p2 = p1 ? p1->next : 0;  
        p3 = p2 ? p2->next : 0;  
    }  
}
```

Initial List: 1, 2, 3, 4, 5, 6, 7, 8, 9

What will be the final contents of the list?

- A 3, 2, 1, 6, 5, 4, 9, 8, 7
- B 1, 2, 3, 4, 5, 6, 7, 8, 9
- C 1, 3, 2, 5, 4, 7, 6, 9, 8
- D 3, 1, 2, 6, 4, 5, 9, 7, 8



Q. Consider a doubly linked list defined as follows:

```
struct Node  
{  
    int Value;  
    struct Node *Fwd;  
    struct Node *Bwd;  
};
```

In this list, Fwd points to the next node and Bwd points to the previous node. Which segment of code correctly deletes the node pointed to by Z, given that Z is neither the head nor the tail of the list?

- A $Z \rightarrow \text{Bwd} \rightarrow \text{Fwd} = Z \rightarrow \text{Fwd}$; $Z \rightarrow \text{Fwd} \rightarrow \text{Bwd} = Z \rightarrow \text{Bwd}$;
- B $Z \rightarrow \text{Bwd} \rightarrow \text{Fwd} = Z$; $Z \rightarrow \text{Fwd} \rightarrow \text{Bwd} = Z$;
- C $Z \rightarrow \text{Bwd} \rightarrow \text{Fwd} = Z \rightarrow \text{Fwd}$; $Z \rightarrow \text{Fwd} \rightarrow \text{Bwd} = Z \rightarrow \text{Fwd}$;
- D $Z \rightarrow \text{Bwd} \rightarrow \text{Fwd} = Z \rightarrow \text{Fwd}$; $Z \rightarrow \text{Fwd} \rightarrow \text{Bwd} = Z \rightarrow \text{Bwd}$;



Q. Consider the following C function that rearranges the elements of a singly linked list. The function is called with a list containing the integers in the given order. What will be the contents of the list after the function completes execution?

```
struct node {  
    int value;  
    struct node *next;  
};  
void rearrange(struct node *list) {  
    struct node *p, *q;  
    int temp;  
    if (!list || !list -> next) return;  
    p = list; q = list -> next;  
    while(q) {  
        if (p->value < q->value) {  
            temp = p -> value;  
            p->value = q -> value;  
            q->value = temp;  
        }  
        p = q;  
        q = q->next;  
    }  
}
```

Initial List: 7, 1, 6, 2, 5, 3, 4

What will be the final contents of the list?

- A 7, 6, 5, 4, 3, 2, 1
- B 7, 1, 6, 2, 5, 3, 4
- C 7, 6, 2, 5, 3, 4, 1
- D 7, 6, 5, 4, 3, 1, 2



- Q. The following C function takes a singly linked list as input and prints its elements iteratively. Some part of the code is left blank.

```
typedef struct node
{
    int value;
    struct node *next;
}
Node;
void printlist(Node *head)
{
    Node* current = head;
    // ---- Blank ----
}
```

Choose the correct alternative to replace the blank line:

- A**

```
while (current != NULL) {
    printf("%d", current->value);
    current = current->next;
}
```
- B**

```
while (current != NULL) {
    printf("%d", current->next->value);
    current = current->next;
}
```
- C**

```
while (current != NULL) {
    current = current->next;
    printf("%d", current->value);
}
```
- D** None of these



Q. The function is called with a list containing the integers in the given order. What will be the contents of the list after the function completes execution?

```
struct node {  
    int value;  
    struct node *next;  
};  
void rearrange(struct node *list) {  
    struct node *p, *q;  
    int temp;  
    if (!list || !list -> next) return;  
    p = list; q = list -> next;  
    while(q) {  
        temp = p -> value;  
        p->value = q -> value;  
        q->value = temp;  
        p = q ->next;  
        q = p ? p ->next : 0;  
    }  
}
```

Initial List: 9, 8, 7, 6, 5, 4, 3, 2, 1

What will be the final contents of the list?

- A 8, 9, 6, 7, 4, 5, 2, 3, 1
- B 9, 7, 8, 5, 6, 3, 4, 1, 2
- C 8, 9, 6, 7, 4, 5, 2, 1, 3
- D 7, 8, 9, 4, 5, 6, 1, 2, 3



THANK YOU!

Here's to a cracking journey ahead!