2 CHAPTER

Arrays

Q.1 Which of the following C expressions access the $(i, j)^{th}$ entry of an (m x n) matrix stored in column major order?

(a) $n \times (i-1) + j$ (b) $m \times (j-1) + i$

(c) $m \times (n-j) + j$ (d) $n \times (m-i) + j$

Q.2 Consider 3 dimensional Array $A[90][30][40]$ stored in linear array in column major order. If the base address starts at 10, what is the location of $A[20][20][30]$? Assume the first element is stored at $A[1][1][1]$.

Q.3 Consider a 3-heap tree which is similar to 2-heap tree. Every node in 3-heap contains maximum of 3-children. If Array is used to store the element of 3-heap, find the children of node i ? Assume the first element of array is at 1.

(a) $3i, 3i+1, 3i+2$ (b) $3i-1, 3i, 3i+1$

(c) $3i+1, 3i+2, 3$ (d) None of these

Q.4 In a compact single dimensional array representation for lower triangular matrices of size $n \times n$, non-zero elements of each row are stored one after another, starting from the first row, the index of the $(i, j)^{th}$ element of the lower triangular matrix in this new representation is

(a) $i + j$ (b) $i + j - 1$

(c) $j + \frac{i(i-1)}{2}$ (d) $i + \frac{j(j-1)}{2}$

Q.5 Consider the following function.

```
int search (int A[ ], int k, int l, int h)
{
    int m;
    if (l == h)
    if (k == A[l]) return l;
        else return -1;
    m = ⌊(l + h)/2⌋;
    if (k ≤ A [m])
        return search (A, k, l, m);
    else
        return search (A, k, m+l, h);
}
```

Above function is implemented to search a key in the sorted array with binary search concept. Find the index of key 15 returned by the above function, if array has the following elements and $l = 0$, $h = 8$ are passed to the function along with array and key.

A	12	14	15	15	15	18	110	120	125
	0	1	2	3	4	5	6	7	8

(a) 2 (b) 3

(c) 4 (d) None of these

Q.6 Consider a two-dimensional array with elements stored in the form of lower triangular matrix. How many elements must be crossed to read $A[4, 2]$ from the array $A[-6,..., +8, -6,..., +8]$ whose base address is 1000? (Assume elements are stored in row major order).

Q.7 Consider the following C code

```
int *P, A[3] = {0, 1, 2};
P = A;
*(P + 2) = 5;
P = A++;
*P = 7;
```

What are the values stored in the array A from index 0 to index 2 after execution of the above code?

(a) 7, 5, 2 (b) 7, 1, 5

(c) 0, 7, 5 (d) None of these

Q.8 Let's look about the algorithm

```
int temp, j, i;
for (i = 1; i < n; i++)
{
    temp = A[i];
```

```
    for (j = i - 1; j ≥ 0 && (A[j] > temp); j - -)
    A [j+1] = A[j];
    A[j] = temp;
}
```

If the array is in reverse sorted order then time complexities will be

(a) $O(n)$ (b) $O(n \log_2 n)$
(c) $O(n^2)$ (d) $O(\log_2 n)$

Q.9 Suppose that we have an array of n data records to sort and that the key of each record has the value 0 or 1. An algorithm for sorting such a set of records require _____ running time.

(a) $O(1)$ (b) $O(n)$
(c) $O(n^2)$ (d) None of these

Q.10 Consider an array A has n-elements in which every element is less than $2n$. What is the running time to check whether the given array has distinct elements?

(a) $O(1)$ (b) $O(n)$
(c) $O(n \log n)$ (d) $O(n^2)$

Q.11 Given an array with both +ve and –ve numbers. Find the two elements such that their sum is closest to zero

Ex.: 60 –10 70 –80 85 gives –80 85

What is the tightest upper bound to solve this problem?

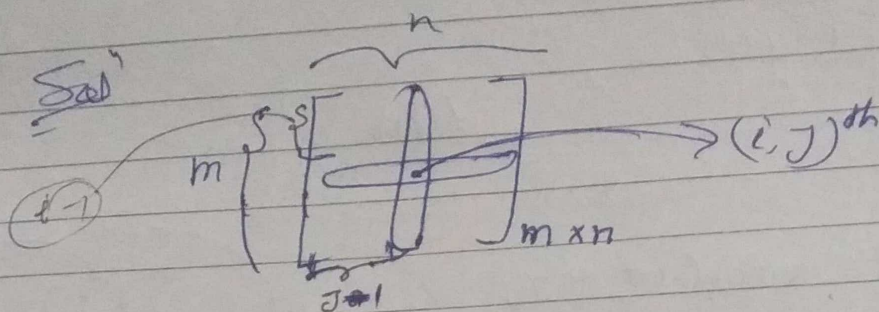
(a) $O(n \log n)$ (b) $O(n^2)$
(c) $O(n^3)$ (d) $O(n)$

■■■■

Array

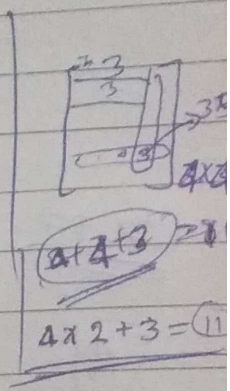
1⇒ $(i, j)^{th}$ of $(m \times n)$ column major order

(B) Soln



$[(j-1) \times (i-1)]$

$m*(j-1)+i$ Ans



Column major

2⇒ BA = 10, A[90][30][40], start = A[1][1][1]

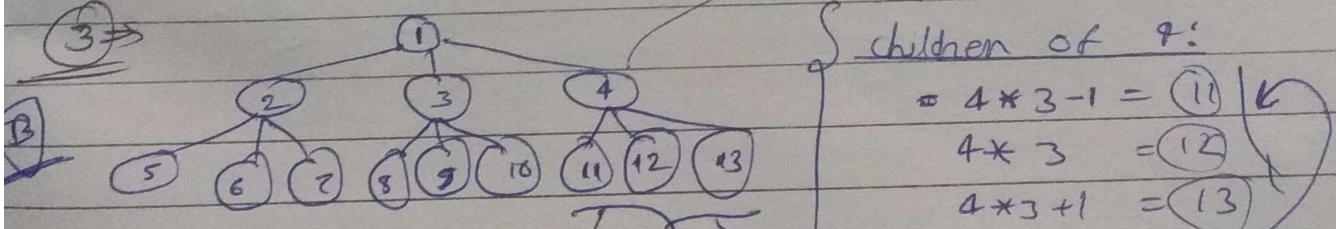
location = A[20][20][30]

Soln

$10 + (20-1)(30)(40) + (30-1)(30) + (20-1)$

$= 23699$ Ans

3⇒



children of 4:

$4*3-1 = 11$

$4*3 = 12$

$4*3+1 = 13$

$(i*3-1), (i*3), (i*3+1)$ Ans

(4) No. of element to be skipped to reach the i^{th} row = $\frac{i(i-1)}{2}$ to reach j^{th} column

(3)

$$= \frac{i(i-1)}{2} + j$$ Ans

12, 13, 15, 15, 15, 18, 110, 120, 125

(15) (0) (8)

(5) int search(int A[], int K, int l, int h)
{

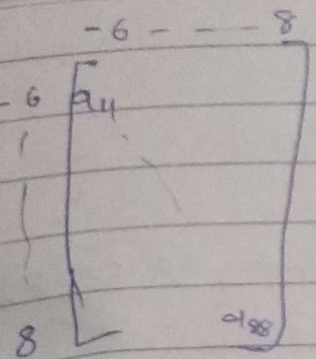
(A)

int m;
if(l==A)
if (K == A[l]) return l;
else return -1;
m = ⌊(l+h)/2⌋;
if (K ≤ A[m])
return search(A, K, l, m);
else
return search(A, K, m+l, h);

}

(15) l == h ⟶ true
K == A[l] ⟶ l = h = 2 Ans

63



63

[l_{b_1} → lower bound of i]

= 1 + 2 + - - - (3 - (-6) + 1)

= 1 + 2 + - - - 10

= 55

[l_{b_2} → lower bound of j]

= 2 - (-6)

= 8

Total = 55 + 8 = 63 Ans

7

```
int *P, A[3] = {0, 1, 2};
P = A;
*(P+2) = 5;
P = A++;
*P = 7;
```

8

Compile Error

Value in A = ?

Error

8)
```
int temp, j, i;
for (i = 1; i < n; i++)
{   temp = A[i];
    for (j = i-1; j >= 0 && (A[j] > temp); j--)
        A[j+1] = A[j];
    A[j] = temp;
}
```

Time complexity = $O(n^2)$ Ans

9) Array → n data records

8) 0 or 1

→ Sort in linear running time

$O(n)$ Ans

10) Array → n elements

8) less than $2n$

→ sorted running time

$O(n)$ Ans

11 ⇒ Array ⟶ +ve & -ve numbers

A

Sum of 2 element is closest to zero

Ex- 60, -10, 70, -80, 85

⟶ -80 & 85

⟶ tightest upper bound = $O(n \log n)$

1. 1st sort element = $n \log n$
2. Add(i) in temp of last & before that set +ve closest = max
   & -ve closest = min

⟶ temp = $a[i] + a[j]$

⟶ $O(n)$

⟶ Total = $O(n \log n) + O(n)$

= $O(n \log n)$ Ans

3 CHAPTER

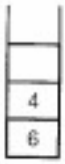
Stack

Q.1 Computes the postfix equivalent of following expression

$$3 * \log(x+1) - \frac{a}{2}$$

(a) 3x1 + log*a2/- (b) 31x + log * a2/-
(c) 31x log + *a2/- (d) 31 logx + * a2/-

Q.2 In evaluating the arithmetic expression 2 * 3 – (4 + 5), using stack to evaluate.
Which of the following is stack configuration is not possible

(a) 

(b) 

(c) 

(d) 

Q.3 Consider the following elements;

1, 2, 3, 4, 5, 6, 7

Use the Binary Max heap to construct a tree with above elements from 1 to 7. Find the children of root.

(a) 5, 6 (b) 4, 6
(c) 3, 6 (d) 2, 6

Q.4 A queue is implemented using two stacks *A* and *B*. Consider the following code.

```
void enqueue (int value)
{
    while (!B.isEmpty ( ))
    A.Push (B.Pop ( ));
    A.Push (value);
}
int dequeue( )
{   while (! A.isEmpty ( ))
    [X]
    return B.Pop( );
}
```

If enqueue is implemented using two stacks *A* and *B* with operations Push, Pop and isEmpty of stack then find the missing statement *X* to implement dequeue operation of queue.

(a) A.Push (B.Pop( ));
(b) B.Push (A.Pop( ));
(c) A.Pop (B.Push( ));
(d) B.Pop (A.Push( ));

Q.5 Consider the following code. Assume initially stack is empty.

```
ch = getchar( );
while (ch != '\n')
{   if ((ch != '+') && (ch != '-') && (ch != '*') &&
    (ch != '/'))
    {  E = atoi (ch); /*converts character to
       integer*/
       push (E);
    }
    else
    {  x1 = pop( );
       x2 = pop( );
       switch (ch)
       {  case '+' : E = x1 + x2; break;
          case '-' : E = x1 - x2; break;
          case '*' : E = x1 * x2; break;
          case '/' : E = x1 / x2; break;
          default : printf ("Not a valid character");
          break;
       }
       push (E);
    }
    ch = getchar ( );
}
```

Assume +, –, * and / operators are used in the expression, the variable ch can hold either operator or integer, x_1 and x_2 are integer variables. Find the content of the stack after executing the program, if the input is combination of operators and integers?

(a) Value of infix expression
(b) Value of prefix expression
(c) Value of postfix expression
(d) None of these

Q.6 Consider the following infix expression of C.

$g + a - t + e * e / x * a / m - r + e - s + u - l + t$

Find the length of substring in the equivalent postfix expression for the above infix and that substring has only operators, and the number of distinct operators and the length of substring is maximum compared to any other substring.

Q.7 Find postfix expression for the following infix expression? Assume ↑ as the highest precedence and follows right to left associativity.

Infix: $(a + b) \uparrow (p + q) \uparrow (r * s * t)$

(a) $ab + pq + \uparrow rs * t * \uparrow$
(b) $ab + pq + \uparrow\uparrow rs * t *$
(c) $ab + pq + rs * t * \uparrow\uparrow$
(d) $ab + pq + rst * * \uparrow\uparrow$

Q.8 Assume ↑ is power operator and it has the highest precedence and follows right associativity. What is the output of the following postfix expression evaluated using operand stack?

(*Note*: 2 ↑ 3 = 8)

9 9 1 * 9 1 ↑ / + 9 – 9 +

(a) 10 (b) 19
(c) 18 (d) 81

Q.9 Consider the implementation of multiple stacks in single array S of size P from index 0 to $P-1$. Size of each stack is Q. The following function PUSH ( ), used to push data x on to a particular stack i where T_i is used as top variable for stack i (i indicates stack number).

```
PUSH (S, P, Q, T_i, x)
{
    if (_____A_____)
    {
        printf ("stack overflow");
        exit (1);
    }
    else
        T_i++;
    S[T_i] = x;
}
```

Stack 0 stores elements from 0 to $Q-1$, stack 1 stores from q to $2Q-1$, and similarly other stack will store elements. Which of the following is the correct expression to replace A in the above function?

(a) $T_i == \left(\frac{P}{Q} \times i - 1\right)$

(b) $T_i == \left(\frac{P}{Q} \times i + 1\right)$

(c) $T_i == \left(\frac{P}{Q} \times (i-1) - 1\right)$

(d) $T_i == \left(\frac{P}{Q} \times (i+1) - 1\right)$

Q.10 Assume stack A has the entries p, q and r (with p on top and r on bottom). Initially stack B is empty. An entry popped out of stack A can be printed immediately or pushed to stack B. A entry popped out of stack B can only be printed.

What is the least number of stack permutations of input sequence that start with a particular letter?

Q.11 The post fix form of $A \$ B * C - D + E / F / (G + H)$ is

(a) $AB \$ C * D - EF / GH + / +$
(b) $AB \$ * C - D + EF / GH / +$
(c) $AB \$ C + D - EF / GH - / +$
(d) $AB \$ C - D * EF / GH / ++$

Q.12 The prefix form of $A - B / (C * D \$ E)$ is

(a) $-/* AB*(C - D)$ (b) $- ABCD* DE\$$
(c) $-A / B*C \$ DE$ (d) $- A / BC*\$ DE$

Q.13 Assume ↑ is power operator and it has the highest precedence and follows right associativity. What is the output of the following postfix expression evaluated using operand stack?

(*Note:* 2 ↑ 3 = 8)

7 7 1 * 7 1 ↑ / + 7 – 7 +

Q.14 Let S be a stack of size $4 \geq 1$ and it is initially empty. Suppose we push the numbers 1, 2, 3, 4 in sequence and then perform 4 pop operations. Let one push operation takes 5 ns; one pop operation takes 5 ns; the time between the end of one such stack operation and the start of the next operation is 2 ns. The stack-life of a particular element $p \geq 1$ is defined as the time elapsed from the end of push (p) to the start of the pop operation that removes p from the stack. The average stack-life of an element of this stack is_____ (in ns).

Q.15 Consider the following infix expression which is to be converted to postfix expression using stack.

$(((P + Q) * (R + S)) / T) + (A * (B + C))$

The sum of all unique possible heights of stack when converting from infix to postfix is _____.

■■■■

Stack

(1) → (A)

postfix = $3 * \log(x+1) - a/2$

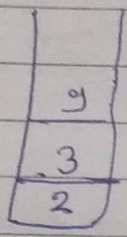
$= 3 * \log(x+1) - a2/$

$= 3\log(x+1) * a2/ -$

$= 3x1 + \log * a2 -$ Ans

(2) → (D)

$2 * 3 - (4+5)$ ⟶ Stack not possible

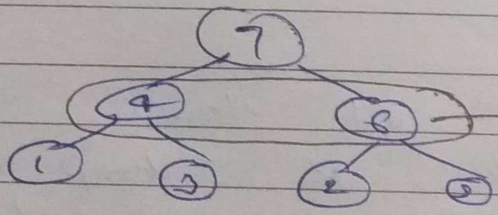


⟹ not possible

(3) → (B)

Max-heap — 1, 2, 3, 4, 5, 6, 7

Child of root



child (4, 6)

(4) → (B)

Queue implement using stack A & stack B

```
void enqueue (int value)
{ while (!B.isEmpty())
      A.push(B.pop());
  A.push(value);
}
```

```
int dequeue()
{ while (!A.isEmpty())
      B.push(A.pop());
  return B.pop();
}
```

5 x 4 → 2

Ans

(5) +, -, *, / ⟶ expression

ch ⟶ either operator or integer

x_1, x_2 ⟶ integer

(a)

→ execution of code

→ Value of postfix expression Ans

(6) g+a-t+e*e/x*a/m-r+e-s+u-l+t

(2)

→ length of substring (operator) in postfix

→ ga+l-ee*x/a*m/+,r-e+s-u+l-t+

2 length operator

length = 2 Ans

(7) (a+b)↑(p+q)↑(r*s*t) ⟶ postfix

(x)

= (ab+)↑(pq+)↑((rs*)*t)

= (ab+)↑(pq+rs**t*)↑

= (ab+pq+rs**t*)↑↑

= (ab+pq+rs**t*↑↑) Ans

(8) $9\,9\,1\,*\,9\,1\,\uparrow\,/\,+\,9\,-\,9\,+$

$= 9\,9\,9\,1\,\uparrow\,/\,+\,9\,-\,9\,+$

$= 9\,9\,9\,/\,+\,9\,-\,9\,+$

$= 9\,1\,+\,9\,-\,9\,+$

$= 10\,9\,-\,9\,+$

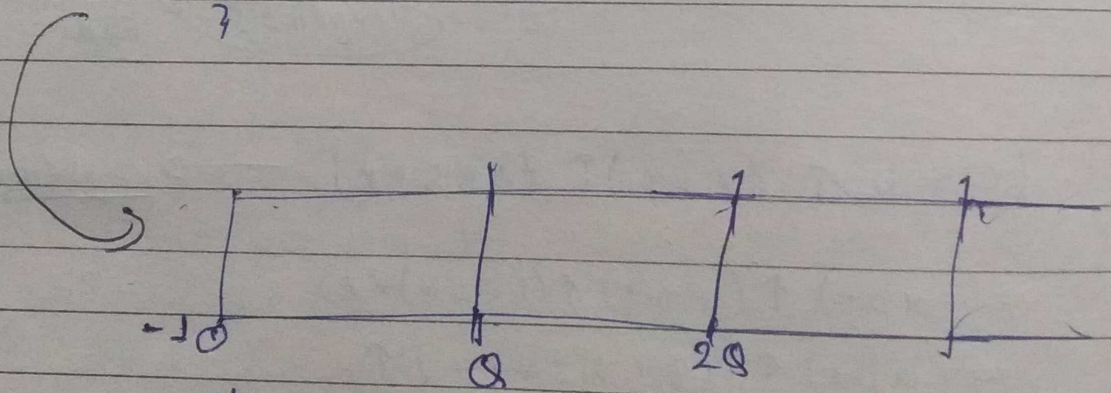
$= 1\,9\,+$

$= 10$ Ans

(9)

(A)

```
PUSH(S, P, Q, Ti, x)
{
    if ( A )
    { printf("Stack Overflow");
      exit(1);
    }
    else
        Ti++;
    S[Ti] = x;
}
```

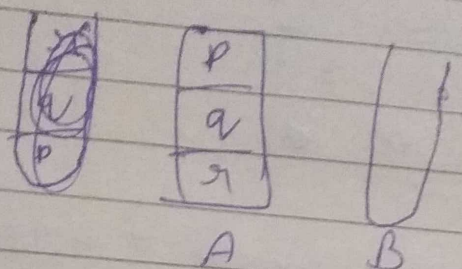


1st element of next stack $= T_{i+1} = \frac{P}{Q}(i-1)$

initial value of stack pointer $= \frac{P}{Q} * i - 1$

Overflow (A) $= T_i = \frac{P}{Q}(i-1) - 1$ Ans

10 → 1



p q r
q p r
q r p
~~p q r~~ p q r
r p q → Least no. of permutation starting with (r)

No. = (1) Ans

11 → A

A $ B * C - D + E / F / (G + H) → Postfix = ?

A B $ C * D - E F / G H + / + Ans

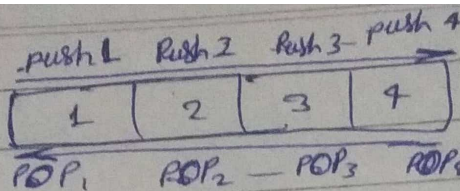
12 → C

A - B / (C * D $ E) → Prefix = ?
~~Postfix~~ = ?

- A / B * C $ D E Ans

13 → 8

7 7 1 * 7 1 ↑ / + 7 - 7 +
= 7 7 7 1 ↑ / + 7 - 7 +
= 7 7 7 / + 7 - 7 +
= 7 1 + 7 - 7 +
= 8 7 - 7 +
= 1 7 +
= 8 Ans



Time for 1 stack operation = 5 ns

stack life in element - 1

= 5ns × 3 + 5ns × 3 + 2ns × 7

= 44 ns

element - 2

= 5ns × (2 × 2) + 2ns × 5

= 30ns

element - 3

= 5ns × (1+1) + 2ns × 3

= 16 ns

element - 4

= 5ns × (0+0) + 2ns × 1

= 2ns

Average stack life

$$= \frac{44 + 30 + 16 + 2}{4}$$

= 23 ns Ans

15 ⇒

(((P+Q)*(R+S))/T)+(A*(B+C)) → Postfix

15

↓

PQ+RS+*T/ABC+*+

Sum of all ~~possible~~ unique possible heights

= 1+2+3+9+5

= 15 Ans