POSTAL Book Package 2021

Computer Science & IT

Objective Practice Sets

Programming and Data Structure

Contents

MADE EASY Publications

2 CHAPTER

Arrays

Q.1 Which of the following C expressions access the $(i, j)^{th}$ entry of an $(m \times n)$ 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) $O(n)$

Q.12 What is the output of the following C code? Assume that the address of X is 2000 (in decimal) and an integer requires four bytes of memory.

```
int main ().
{
unsigned int × [4] [3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}}
printf (“%u,%u,%u”, X+3, ∗ (X + 2) + 3);
}
```

(a) 2036, 2036, 2036
(b) 2012, 4, 2204
(c) 2036, 10, 10
(d) 2012, 4, 6

Q.13 Consider the following C function:

```
#include <stdio.h>
int main(void)
{
    char c[ ] = “ICRBCSIT17”;
    char *p = c;
    printf(“%s”, c+2[p] – 6[p] – 1);
    return 0;
}
```

The output of the program is

(a) SI (b) IT
(c) T1 (d) 17

Q.14 The output of the following program is

```
main ( )
{   static int x[ ] = {1, 2, 3, 4, 5, 6, 7, 8};
    int i ;
    for (i = 2; i < 6; ++i)
    x[x[i]] = x[i];
    for (i = 0; i < 8; ++i)
    printf(“%d”, x[i]);
}
```

(a) 1 2 3 3 5 5 7 8 (b) 1 2 3 4 5 6 7 8
(c) 8 7 6 5 4 3 2 1 (d) 1 2 3 5 4 6 7 8

Q.15 Which of the following is true?

(a) In sorted array of ‘n’ distinct elements, deletion of an element from beginning takes $O(\log n)$ time.
(b) In sorted array of ‘n’ distinct elements, insertion of an element takes $O(\log n)$ time.
(c) In sorted array of ‘n’ distinct elements, finding i^{th} largest element take $O(1)$ time.
(d) In unsorted array of ‘n’ distinct elements, insertion of an element take $\Omega(\log n)$ time.

Q.16 Consider the function given below, which should return the index of first zero in input array of length ‘n’ if present else return –1.

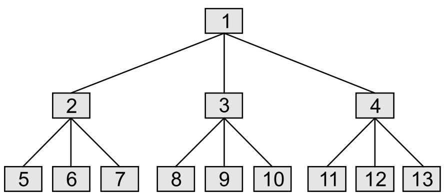
```
int index of zero (int[ ] array, int n) {
for (int i = 0; [P] ; i++);
    if (i = = n)
        return –1;
    return i;
}
```

For column major order

loc $(A(i, j, k))$ = Base Address $+(i-1)r_2r_3+(K-1)r_2+(j-1)$

$= 10 + 19 * (30)(40) + 29 * (30) + 19 = 23699$

3. (b)



Children of 4:

$4 * 3 - 1 = 11$

$4 * 3 = 12$

$4 * 3 + 1 = 13$

$\therefore$ $(i * 3) - 1$, $i * 3$ and $(i * 3) + 1$ are children of i.

4. (c)

The number of elements to be skipped to reach to i^{th} row $= \frac{i(i-1)}{2}$ to reach to j^{th} column $= \frac{i(i-1)}{2} + j$.

5. (a)

(ℓ)				(m)				(h)
12	14	15	15	15	18	110	120	125

(ℓ)		(m)		(h)
12	14	15	15	15

(ℓ)	(m)	(h)
12	14	15

$15 \Rightarrow l == h$ is true and $k == A[l]$.

$[l = h = 2]$

$\therefore$ Option (a) is correct.

6. (63)

The given lower triangular matrix can be represented as

$$\begin{array}{c|cccccc|} & -6 & -5 & -4 & \ldots & & +8 \\ \hline -6 & a_{11} & & & & & \\ -5 & a_{21} & a_{22} & & & & \\ -4 & a_{31} & a_{32} & a_{33} & & & \\ \cdot & \cdot & \cdot & & & & \\ \cdot & \cdot & \cdot & & & & \\ \cdot & \cdot & \cdot & & & & \\ \cdot & \cdot & \cdot & & & & \\ +8 & a_{81} & a_{82} & \ldots & & & a_{88} \end{array}$$

Let (i, j) be the element to be accessed.

We must cross upto $(i-1)^{th}$ row.

Number of elements upto $(i-1)^{th}$ row or 10^{th} row

$= 1 + 2 + 3 + ... + [(i-1) - (l_{bi}) + 1]$

$[l_{bi} \rightarrow$ lower bound of $i]$

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

$= 1 + 2 + 3 + ... + (10)$

$$= \frac{10 \times 11}{2} = 55$$

In i^{th} row we must cross $(j - l_{bj})$ elements.

$[l_{bj} \rightarrow$ lower bound of $j]$

$= 2 - (-6) = 8$

$\therefore$ In total $= 55 + 8 = 63$ elements need to be crossed.

7. (d)

$P = A{+}{+}$; produces compiler error.

So execution of the given code is not possible. $A{+}{+}$ asks the compiler to change the base address of an array, but compiler knows A is array hence once it is declared, compiler will not allow to change the address.

8. (c)

In this programme first for loop will run n-times and second for loop also run n-times, because our array is reverse sorted then second loop will also run and the total time complexity for reverse sorted order will be $O(n^2)$.

9. (b)

Using counting sort, it takes linear time.

10. (b)

Using counting sort, single scan will identify if there exist any repeated element in the given array. Therefore, it takes $O(n)$ time.

11. (a)

1. First sort elements $\rightarrow n \log n$.
2. Add (i) and in temp at last and before that set +ve closest = max and –ve closest-min

```
temp = A[i] + A[j]
if (temp > 0)
{
    if (temp < positive closest)
    positive closest = temp;
```