Ref. No. : Ex/IEE/PC/B/IT/T/221/2024

B.E. INSTRUMENTATION AND ELECTRONICS ENGINEERING SECOND YEAR SECOND

EVEN SEMESTER EXAM - 2023

DATA STRUCTURE, ALGORITHM & OOPS

Time : 3 hrs **Full Marks :100**

Section –I 2*10=20

1. What is a tree?

2. What condition leads to worst case in quick sort? What is the expression of time complexity in this case?

3. Out of the algorithms quick sort, heap sort and merge sort which performs best in worst case?

4. Which of the sorting algorithm require minimum number of swaps?

5. Consider the following C++ code segment.

```
public int IsPrime(int n){
   for(int i = 2 ; i <= sqrt(n);i++){
      if(n % i == 0){
         printf("Not Prime");
         return 0;
      }
      return 1;
   }
}
```

Write the expression of complexity in terms of O.

6. What are the components of hashing?

7. What is Starsseien matrix multiplication? What is the advantage of it over normal matrix multiplication?

8. Explain the C++ code syntax given below

```
Shape(double w = 0.0, double h = 0.0) : width(w), height(h) {}
```

9. The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k)=k mod 10 and linear probing. What is the resultant hash table?

10. Solve the following recurrence relation using master theorem -

$$T(n) = 4T(n/2) + n^2$$

[Turn over

Section II **4*5=20 Marks**

(Any four)

1. Given the following list of numbers [14, 17, 13, 15, 19, 10, 3, 16, 9, 12] find the contents of the list after the second partitioning according to the quicksort algorithm?
2. What are the advantages of dynamic programing over recursive one, explain with an example?
3. What are practical application of stack , linked lists?
4. Using master theorem find the complexity of T(n) = 2^n T(n/2) + n^n
5. Given an array A of n elements, each of which is an integer in the range [1, n^2], how do we sort the array in O(n) time?

Section III **2*10=20 Marks**

1.

Find the time complexity of the lines of code

i). void function(int n)

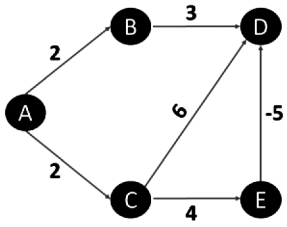
```
{
        int i = 1, s = 1;
        while (s < n) {
                s = s + i;
                i++;
        }
}
```

ii)

```
fib (n)
{
  if n <= 1:
     return 1;
  return fib(n-1) + fib(n-2);
}.
```

(5 Marl‹s)

1. (b) Using bellman ford algorithm show step by step the shortest path between A to E (include the in between steps)



. (5 Marks),

2.

(a) Determine and explain the functionality of the following code fragment: (2 Marks)

```
int functionl (int a[], int n, int x)
{
 int i;
for (i = 0; i < n && a[i] != x; i++);
 if (i == n)
return -1;
else
return i;
}
```

(b) Compile the best case, worst case and average case time complexity of the above functionality. Explain your answers. (3 Marks)

(c) Determine and explain the functionality of the following code fragment: (2 Marks)

```
 int function2 (int a[], int n, int x)
{
int i,j,k;
 i = 0; j = n -1;
while (i <= j)
{
k = (i + j)/2;
if (x==  a[k])  return k;
if (x >= a[k])
 i=k+1;
else
j = k - 1 ;
}
return -1;
}
```

(d) Compute the best case, worst case and average case time complexity of the function2. Explain your answers. (3 Marks)

Section IV **2*20=40**

1.

a. Explain the visibility of base class members for the access specifiers: private, protected, and public while creating the derived class and also explain the syntax for creating derived class. (5 marks)
b. Explain with suitable examples, multilevel inheritance and multiple inheritances. (5 marks)
c. Define pure virtual functions. write a C++ program to illustrate pure virtual functions (5 marks)
d. What do you mean by function overloading in C++. Write a C++ program to add two integer, double, strings, using operator overloading (+) operator . (5 marks)

2.

a. State the important features of OOPS. Compare OOPS with procedure oriented programing. (5 marks)
b. Using virtual function access the protected members of stack data structure to implement the stack function push and pop (5 marks)
c. Write a C++ program to illustrate the concept of abstraction, encapsulation, polymorphism (5 marks)
d. Write a C++ program to calculate the average value of the stack (using an array) elements. (5 marks)