2223476802 3
Satyandra Pal
Neelima Mom

Master of Computer Applications

MCAC-201: Design and Analysis of Algorithms

Unique Paper Code: 223421201

Semester II

August-2023

Year of Admission: 2022

Duration: 3 hours | Max Marks: 70

Note: Attempt all questions. Write neatly and absolutely to the point. Assumptions (if any) should be clearly stated.

1. Given an array $arr[]$ with n distinct integers; give an $\mathcal{O}(1)$ time algorithm to return an element which is neither the maximum nor the minimum in this array. (3)

2. An array of 10 distinct elements is to be sorted using Quick Sort. Assume that the pivot element is chosen uniformly at random. What is the probability that the pivot element gets placed in the worst possible location in the first round of partitioning? (3)

3. Give the worst case time complexity to sort n^2 elements using (3*2=6)
   (a) Insertion Sort
   (b) Merge Sort
   (c) Quick Sort

4. The first two characters of an Indian Registration Number (Number Plates) are letters denoting the federal state or union territory, of which there are 36 in India. Next up are two digits indicating the regional transport authority within the state or territory that issued the registration number. After that, there are two letters showing the current series of registration numbers. The final four digits are the number unique to the plate. For example for $DL07CR5077$, DL represent the union territory of Delhi, 07 is the regional transport authority number, CR denote the series and 5077 is the unique number.
   Suppose we have Number Plate records for n people, and we wish to sort them based on their Number Plate Registration Number. Give a linear time algorithm to do the same. (5)

5. Given an array $arr[]$ with n distinct integers; already sorted in **decreasing order**, give an $\mathcal{O}(lgn)$ time algorithm to return the index of the key ele in $arr[]$ if present and -1 if not. (5)

6. Given an array $arr[]$ and a key to be searched ele, Consider a divide and conquer algorithm as follows:
   Break the array into two equal subarrays. Search for ele in each of the subarrays. Return True if ele is found in either of the two subarrays, else False.
   Give the recurrence relation for worst case time complexity the above algorithm. Also analyze the worst case running time of the above algorithm. (5)

7. Given k different sorted arrays $A_1, A_2, \ldots A_k$ each containing n integers, give an efficient algorithm to merge these k sorted arrays into a single sorted array. Analyze the time complexity of your algorithm.

p_i rupees (i is an integer).
Give a Dynamic Programming based algorithm to solve rod cutting problem. Analyze the time and space complexity of your algorithm. (8)

9. Convert the following algorithm into an efficient Dynamic Programming based solution. Also compare the time and space complexity of the original algorithm to your Dynamic Programming based solution. (7)

```
fibonacci(num)
    if num = 0
        return 0
    elseif num = 1
        return 1
    else
        return fibonacci(num-1)+ fibonacci(num-2)
```

10. Consider *Dijkstra's* Algorithm for shortest path problem. (4+4=8)
    (a) Does it produce an optimal solution if we allow some edge weights to be zero? (Edge weight can be either positive or zero). Justify.
    (b) Does it produce an optimal solution if we allow some edge weights to be negative? (Edge weight can be either positive or negative). Justify.

11. The following statements may or may not be correct. In each case, either prove it (if it is correct) or give a counter example/argument (if it is not correct). (5*3=15)
    (a) For an undirected weighted graph $G(V, E)$ such that each edge weight is distinct. Then for a given pair of vertices $u, v \in V$, there is a unique shortest path from u to v. (Assume all weight are strictly positive).
    (b) For an undirected weighted graph $G(V, E)$ such that each edge weight is distinct. If G contains atleast one cycle, then the maximim weight edge (lets says (u, v)) cannot be present into any minimum spanning tree of G. (Assume all weight are strictly positive).
    (c) Let A_1 and A_2 be two algorithms to perform some task. The worst case running time of A_1 is $Theta(n.lgn)$ whereas the worst case running time of A_2 is $Theta(n^2)$. Then there must be an input instance i for which A_1 performs asymptotically better than A_2.
    (d) Let A_1 and A_2 be two algorithms to perform some task. The worst case running time of A_1 is $Theta(n.lgn)$ whereas the worst case running time of A_2 is $Theta(n^2)$. Then for every input instance, A_1 performs asymptotically better than A_2.
    (e) Given a sorted array $arr[]$ with n integers, and a key ele. We can compute the frequency of ele in $arr[]$ is $\mathcal{O}(lgn)$ time.

Suppose you have a rod of length n,and you want to cut up the rod and sell the pieces in a way that maximizes the total amount of money you get. A piece of length i is worth p_i rupees (i is an integer).
Give a Dynamic Programming based algorithm to solve rod cutting problem. Analyze the time and space complexity of your algorithm. (8)

9. Convert the following algorithm into an efficient Dynamic Programming based solution. Also compare the time and space complexity of the original algorithm to your Dynamic Programming based solution. (7)

```
fibonacci(num)
    if num = 0
        return 0
    elseif num = 1
        return 1
    else
        return fibonacci(num−1)+ fibonacci(num−2)
```

10. Consider *Dijkstra's* Algorithm for shortest path problem. (4+4=8)

(a) Does it produce an optimal solution if we allow some edge weights to be zero? (Edge weight can be either positive or zero). Justify.

(b) Does it produce an optimal solution if we allow some edge weights to be negative? (Edge weight can be either positive or negative). Justify.

11. The following statements may or may not be correct. In each case, either prove it (if it is correct) or give a counter example/argument (if it is not correct). (5*3=15)

(a) For an undirected weighted graph $G(V, E)$ such that each edge weight is distinct. Then for a given pair of vertices $u, v \in V$, there is a unique shortest path from u to v. (Assume all weight are strictly positive).

(b) For an undirected weighted graph $G(V, E)$ such that each edge weight is distinct. If G contains atleast one cycle, then the maximim weight edge (lets says (u, v)) cannot be present into any minimum spanning tree of G. (Assume all weight are strictly positive).

(c) Let A_1 and A_2 be two algorithms to perform some task. The worst case running time of A_1 is $Theta(n.lgn)$ whereas the worst case running time of A_2 is $Theta(n^2)$. Then there must be an input instance i for which A_1 performs asymptotically better than A_2.

(d) Let A_1 and A_2 be two algorithms to perform some task. The worst case running time of A_1 is $Theta(n.lgn)$ whereas the worst case running time of A_2 is $Theta(n^2)$. Then for every input instance, A_1 performs asymptotically better than A_2.

(e) Given a sorted array $arr[]$ with n integers, and a key ele. We can compute the frequency of ele in $arr[]$ is $\mathcal{O}(lgn)$ time.