Office Copy
F-First
2

Roll Number:____________________ Number of Pages: 03

Thapar Institute of Engineering and Technology, Patiala

Department of Computer Science and Engineering

END SEMESTER EXAMINATION

B. E. (Second Year): Semester-II (2018/19)	Course Code: UCS406 Course Name: Data Structures and Algorithms
May 11, 2019	Saturday, 09:00 Hrs – 12:00 Hrs
Time: 3 Hours, M. Marks: 100	Name of Faculty: SMG, SUG, SP, TBH, RKR, RAH, ASG, ANK

Note: Attempt all questions (sub-parts) in sequence. Assume missing data, if any, suitably.

Q1. Consider a hash table of size $m = 7$ and a corresponding hash function $h(k) = k \bmod m$. Compute the locations to which the keys $\mathbf{99, 59, 26, 50,}$ and $\mathbf{58}$ are mapped using the following collision resolution techniques.

(a) Chaining. (5)

(b) Quadratic probing. (5)

Q2. (a) Draw a Binary Search Tree by sequentially inserting the following elements: (4)

$$\mathbf{40, 50, 22, 33, 30, 80, 15, 25, 60, 90, 75, 44}$$

(b) For the BST obtained in **Q2.(a)**, print the elements in Pre-order, In-order and Post-order traversals. (3)

(c) Delete in sequence 44, 50, and 40 from BST obtained in **Q2.(a)**. Show the BST after each deletion. (3)

Q3. Define P and NP Complete class of problems. Give at-least one example for each class. (4)

Q4. Show stepwise a tree that Huffman's greedy algorithm could produce for the sentence given in **Fig. 1.** Write optimal codes thus generated for all the different characters present in the sentence (**Fig. 1**). (12)

DATASTRUCTURESTRUCTURESDATA

Fig. 1

Note: Consider only those characters which are present in the given sentence.

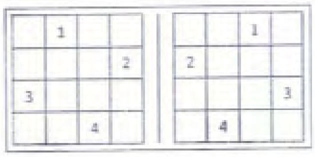
Q5. Use dynamic programming to fully parenthesize the product of four matrices, i.e. $\boldsymbol{A}\,[10 \times 5], \boldsymbol{B}\,[5 \times 20], \boldsymbol{C}\,[20 \times 10], \boldsymbol{D}\,[10 \times 5]$, such that the number of scalar multiplications gets minimized. Show each and every step. (10)

Q6. Illustrate stepwise execution of the Heapsort algorithm on an array $\boldsymbol{A} = [\mathbf{44, 30, 50, 22, 60, 55, 77, 55}]$ to arrange its elements in descending order. (10)

Q7. In n-Queens, some solutions are simply reflections of others. For example, two solutions (**Fig. 2**) for 4-Queens problem are equivalent under reflection. **(10)**

(a) Modify NQueens Algorithm (**Fig. 3**) to get unique solutions only.

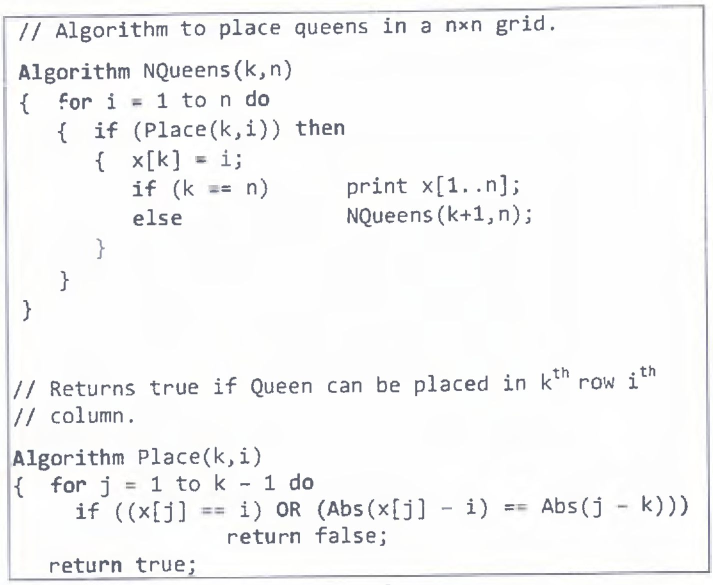
(b) Execute the NQueens Algorithms devised in **Q7. (a)** for $n = 1, 2, 3,$ and 4. Draw the associated state space search trees separately for each value of n.



	1		
			2
3			
		4	

		1	
2			
			3
	4		

Fig. 2



```
// Algorithm to place queens in a nxn grid.

Algorithm NQueens(k,n)
{  for i = 1 to n do
   {  if (Place(k,i)) then
      {  x[k] = i;
         if (k == n)      print x[1..n];
         else             NQueens(k+1,n);
      }
   }
}


// Returns true if Queen can be placed in kth row ith
// column.

Algorithm Place(k,i)
{  for j = 1 to k - 1 do
     if ((x[j] == i) OR (Abs(x[j] - i) == Abs(j - k)))
                  return false;
   return true;
```

Fig. 3

Q8. (a) Run Dijkstra's algorithm on the directed graph (**Fig. 4**), starting at vertex S. Show all the intermediate graphs in deriving the final shortest path tree. What is the order in which vertices get removed from the priority queue? **(5)**

(b) Write sequence in which nodes of the graph (**Fig. 5**) have been traversed using DFS and BFS, starting at vertex A. To make a unique solution, assume that whenever you faced with a decision of which node to pick from a set of nodes, pick the node whose label occurs earliest in the alphabet. **(5)**

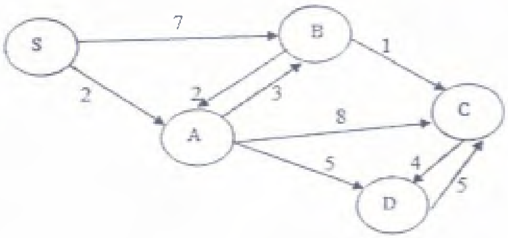


Fig. 4

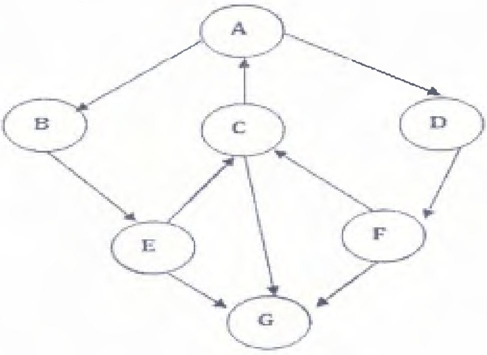


Fig. 5

Q9. (a) Let each node x in a binary search tree keeps an attribute $x.successor$ that points to x's inorder successor. Give a pseudocode for **INSERT** on a binary search tree T using this representation. **(6)**

(b) Given two integer arrays $a[1..m]$ and $b[1..n]$, find an integer that appears in both arrays (or report that no such integer exists). Assuming $m \leq n$, give a crisp and concise algorithm in structured English satisfying following performance requirements: **(6)**

- The amount of extra space (besides $a[\,]$ and $b[\,]$) must be constant. It is fine to modify $a[\,]$ and $b[\,]$.
- The worst case running time must be $O(n \log m)$.

Note: Algorithm will be evaluated on correctness, efficiency, and clarity.

Q10. (a) **Fig. 6** presents an algorithm to find the minimum in a stack of integers without affecting its contents. Determine if there is any bug in this code with respect to the following. Give suitable examples for each justification. **(6)**

i. Change in stack contents.

ii. Incorrect answer.

```
// S and temp are stacks. S contains some integers whose minimum is to be
// determined. isEmpty(S) returns true if S is empty, else returns false.
// min and t are integer variables.
       if (!isEmpty(S))
       {  min = pop(S);
          while (!isEmpty(S))
          {  t = pop(S);
             if (t < min)
                 min = t;
             push(temp, t);
          }
          while (!isEmpty(temp))
             push(S,pop(temp));
          return min;
       }
```

Fig. 6

(b) Consider a recursive static function $f()$, given below to answer the following questions **(6)**

```
static int f(int n)
{ if (n == 0) return 0;
  if (n == 1) return 0;
  if (n == 2) return 1;
  return f(n - 1) + f(n - 2) - f(n - 3);
}
```

i. What is the value of $f(3)$?
ii. What is the value of $f(4)$?
iii. What is the value of $f(7)$?
iv. How many calls on $f()$ are made to compute $f(7)$, including the first one?
v. What is the value of $f(\mathbf{101})$?

---**ALL THE BEST**---