Data Structures & Algorithms (CS F211) Comprehensive Exam
Part A (Closed Book)

There are 20 questions in Part A with total marks $40 = 20 \times 2$. You will get Part B (Open Book) only after you have submitted your Part A question paper (with answers written on it). You can use the answer sheet provided with this question paper and blank space in the question paper for rough work. Time: (Part A) + (Part B) = 180 minutes.

**Please write your name, ID, and signature here:**
Name:
ID:
Signature:
Please write all your answers (A, B, C, or D) here:

Question#	1	2	3	4	5	6	7	8	9	10
Answer										
Question#	11	12	13	14	15	16	17	18	19	20
Answer										

1. If there are n integers to sort, each integer has d digits and each digit is in the set $\{1, 2, ..., k\}$, radix sort can sort the numbers in time:

   (A) $O(dnk)$

   (B) $O(dn^k)$

   (C) $O((d+n)k)$

   (D) $O(d(n+k))$

2. The solution of the recurrence relation $T(n) = 3T(n/4) + \sqrt{n}$ is:

   (A) $\Theta(n^{\log_3 4})$

   (B) $\Theta(n^{\log_4 3})$

   (C) $\Theta(n\sqrt{n})$

   (D) $\Theta(n\log(n))$

3. Floyd-Warshall algorithm utilizes ——— to solve the all-pairs shortest paths problem on a directed graph in ——— time.

   (A) Greedy Algorithm, $\Theta(V^3)$

   (B) Greedy Algorithm, $\Theta(V^2 \log(V))$

   (C) Dynamic Programming, $\Theta(V^3)$

   (D) Dynamic Programming, $\Theta(V^2 \log(V))$

4. You have an array of n elements. Suppose you implement QuickSort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is:

   (A) $O(n^2)$

   (B) $O(n\log(n))$

   (C) $\Theta(n\log(n))$

   (D) $O(n^3)$

5. The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is:

   (A) 110

   (B) 148

   (C) 155

   (D) 147

6. Suppose P, Q, R, S, T are sorted sequences having lengths $20, 24, 30, 35, 50$ respectively. They are to be merged into a single sequence by merging together two sequences at a time. The number of comparisons that will be needed in the worst case by the optimal algorithm for doing this is:

   (A) 350

   (B) 358

   (C) 210

   (D) 362

7. Which one of the following correctly determines the solution of the recurrence relation with $T(1) = 1$?
   $T(n) = 2T(n/2) + \log(n)$

   (A) $\Theta(n)$

   (B) $\Theta(n \log(n))$

   (C) $\Theta(n^2)$

   (D) $\Theta(\log(n))$

8. Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time on Depth First Search of G? Assume that the graph is represented using adjacency matrix.

   (A) $O(n)$

   (B) $O(m + n)$

   (C) $O(n^2)$

   (D) $O(mn)$

9. Suppose we have three algorithms for solving a given problem: algorithm A uses divide and conquer, algorithm B is a greedy algorithm, and algorithm C uses dynamic programming. Let $T(X)$ denote the time complexity of the algorithm X. Which of the following statements is TRUE?

   (A) $T(A) < T(B) < T(C)$

   (B) $T(C) < T(B) < T(A)$

   (C) $T(C) < T(A) < T(B)$

   (D) $T(B) < T(C) < T(A)$

10. Consider the directed graph given below. Which one of the following is TRUE?
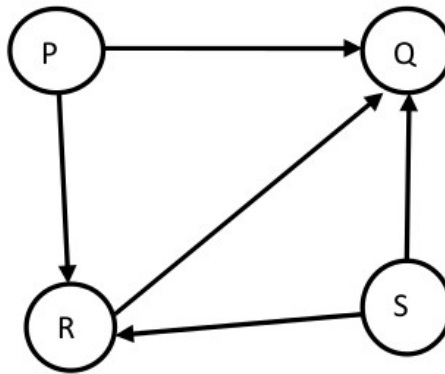
(A) The graph doesn't have any topological ordering

(B) Both PQRS and SRPQ are topological ordering

(C) Both PSRQ and SPRQ are topological ordering

(D) PSRQ is the only topological ordering

11. Complete the following code at line 3 to insert an item x into a list after a node pointed to by p in a singly linked linear list.

```
insertAfter(p, x)
{
  q = getnode();
  info(q) = x;
  /*Line 3*/
  next(p) = q;
}
```

(A) `next(q) = next(p);`

(B) `next(p) = next(q);`

(C) `q = next(p);`

(D) `p = next(q);`

12. Complete the following code at line 3 to delete the node following the node pointed to by p and assign its contents to the variable x in a singly linked linear list.

```
deleteAfter(p, x)
{
  q = next(p);
  x = info(q);
  /*Line 3*/
  freenode(q);
}
```

(A) `next(q) = next(p);`

(B) `next(p) = next(q);`

(C) `q = next(p);`

(D) `p = next(q);`

13. Complete the following code at line 1 so that the function concat(list1, list2) appends the singly linked circular list pointed to by list2 to the end of the singly linked circular list pointed to by list1. The list pointer points to the last node in the circular list.

```
concat(list1, list2)
{
  /*Line 1*/
  next(list1) = next(list2);
  next(list2) = p;
  list1 = list2;
}
```

(A) `next(list1) = next(p);`

(B) `next(p) = next(list1);`

(C) `list1 = next(p);`

(D) `p = next(list1);`

14. Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could NOT be the sequence of nodes examined?

(A) $935, 278, 347, 621, 299, 392, 358, 363$.

(B) $2, 252, 401, 398, 330, 344, 397, 363$.

(C) $924, 220, 911, 244, 898, 258, 362, 363$.

(D) $2, 399, 387, 219, 266, 382, 381, 278, 363$.

15. The time complexity of the Karatsuba's Divide and Conquer Multiplication Algorithm is:

(A) $O(n^{\log_3 2})$

(B) $O(n^{\log_2 3})$

(C) $O(n\sqrt{n})$

(D) $O(n\log(n))$

For questions 16 to 20: Insert the nodes <DEC, NOV, OCT, SEP, AUG, JUL, JUN, MAY, APR, MAR, FEB, JAN> in order from left to right in an empty AVL tree using the dictionary order as the key (if required use rotations to convert back into AVL tree in each step) and answer the following questions.

16. The root node of the final AVL tree is:

(A) DEC

(B) JAN

(C) JUL

(D) JUN

17. The *Average Balance Factor* of the final AVL tree is:

    (A) 0

    (B) 1/6

    (C) 1/12

    (D) −1/12

18. The total number and types of rotations performed to get the final AVL tree is:

    (A) `2RR+2LR`

    (B) `1RR+1LL+1LR+1RL`

    (C) `2LL+2RL`

    (D) `2LR+2RL`

19. *Preorder Traversal* of the final AVL tree is:

    (A) `<APR, AUG, FEB, DEC, JUN, JUL, MAY, SEP, OCT, NOV, MAR, JAN>`

    (B) `<JAN, DEC, AUG, APR, FEB, MAR, JUL, JUN, NOV, MAY, OCT, SEP>`

    (C) `<JUN, DEC, AUG, APR, JAN, FEB, JUL, NOV, MAY, MAR, OCT, SEP>`

    (D) `<APR, AUG, FEB, JUL, JAN, DEC, MAR, MAY, SEP, OCT, NOV, JUN>`

20. *Postorder Traversal* of the final AVL tree is:

    (A) `<APR, AUG, FEB, DEC, JUN, JUL, MAY, SEP, OCT, NOV, MAR, JAN>`

    (B) `<JAN, DEC, AUG, APR, FEB, MAR, JUL, JUN, NOV, MAY, OCT, SEP>`

    (C) `<JUN, DEC, AUG, APR, JAN, FEB, JUL, NOV, MAY, MAR, OCT, SEP>`

    (D) `<APR, AUG, FEB, JUL, JAN, DEC, MAR, MAY, SEP, OCT, NOV, JUN>`