Name: _____ **Application number**: _____

PhD / MS(R) Admission Exam: Sample Duration: 2 hours

This test consists of three sections. You get a total of 50 minutes for the objective questions, 40 minutes for programming problems, and 30 minutes for solving subjective problems. The three parts of the test are to be attempted in the given order.

1 Objective Questions: (96 marks)

1.1 Multiple Choice

Each of the following questions has 4 choices of answer, exactly one of which is correct. You get 4 marks if you pick the correct answer, -1 if you pick the wrong answer, and 0 if you leave the question unattempted.

1. Let w, g and n be propositions where w is "I walk to work", g is "I work in Gurugram", and n is "I work at night". The sentence "When I work at night and I work in Gurugram, I don't walk to work" could be equivalently written using propositions and logical connectives as:

    1. $(n \wedge g) \Rightarrow w$
    2. $(n \vee g) \Leftrightarrow n$
    3. $n \Rightarrow \neg(w \wedge g)$
    4. $\neg(w \wedge g) \vee n$

2. Let $G = (V, E)$ be a connected weighted graph. Which of the following statements is true? (Here, MST means minimum weight spanning tree.)

    1. If G has a unique MST then G must be acyclic.
    2. If G is acyclic then an MST of G can be computed in $O(|V| + |E|)$ time.
    3. For any given cycle C in G, an MST of G can only include at most one edge from C.
    4. G has a unique MST only if deleting any edge in G results in a graph with a larger MST weight.

3. Suppose we have a binary search tree storing a set S of distinct integers and there is a key $k \in S$ stored in a node u in it. The *inorder successor* of u is the node u' containing k' which is the smallest value in S larger than k. Which of the following is true?

    1. The depth of u' is always less than the depth of u.
    2. The subtree rooted at u' always contains u.
    3. The subtree rooted at u may contain u'.
    4. If u doesn't have an inorder successor then u must be a leaf.

4. Suppose E and E' are independent events in a common probability space with probability $1/2$ each. Define events E_1, E_2, E_3 as follows: $E_1 = E \cap E'$, $E_2 = E \cup E'$, $E_3 = E \oplus E' = (E \cap \overline{E'}) \cup (\overline{E} \cap E')$. Which of the following is a pair of independent events?

    1. E and E_1
    2. E_1 and E_2
    3. E and E_3
    4. E_2 and E_3

5. Let A be an $n \times n$ matrix that is not invertible ($n > 1$). Which of the following statements is true?

    1. For every vector b, there exists a vector x such that $Ax = b$.

2. There is exactly one solution to $Ax = 0$.
3. Every vector in $\mathbb{R}^n$ can be written as a linear combination of columns of A.
4. There is a vector b for which $Ax = b$ has more than one solution.

6. Suppose A is an $m \times n$ matrix and B is an $n \times m$ matrix. Which of the following is necessarily true?

   1. If $m \leq n$ then AB is invertible.
   2. If AB is invertible then $m \leq n$.
   3. If $m \geq n$ then AB is invertible.
   4. If AB is invertible then $m \geq n$.

7. For $n \in \mathbb{N}$, let $\ell(n)$ denote the number of ones in the binary representation of n. Which of the following is not true for all $n \in \mathbb{N}$?

   1. $\ell(n) \leq 1 + \log_2 n$.
   2. If n is even, then $\ell(n+1) = \ell(n) + 1$.
   3. If n is odd, then $\ell((n-1)/2) = \ell(n) - 1$.
   4. $\ell(2n) = 2\ell(n)$.

8. The running time $T(n)$ of an algorithm satisfies the recurrence $T(n) = 3T(n/2) + cn$ for some constant c independent of n. Then $T(n)$ is

   1. $\Theta(n^2)$.
   2. $\Theta(n^{\log_2 3})$.
   3. $\Theta(n^{\log_3 2})$.
   4. $\Theta(n^{3/2})$

9. Let X be a real-valued random variable and let μ be its expectation. Which of the following is not necessarily true?

   1. $\mathbb{E}[X - \mu] \geq 0$.
   2. $\mathbb{E}[(X - \mu)^2] \geq 0$.
   3. $\mathbb{E}[(X - \mu)^3] \geq 0$.
   4. $\mathbb{E}[(X - \mu)^4] \geq 0$.

10. A singly linked list contains n integer keys in ascending order, where $n > 10000$. You are given a pointer / reference of the first node of the linked list. Consider the following operations.

    - Find the 1000'th smallest key in the list.
    - Find the 100'th largest key in the list.
    - Determine whether a number k is a key in the list.

    How many of the above operations can be done in $O(\log n)$ time?

    1. 0 (none)
    2. 1
    3. 2
    4. 3 (all)

11. Consider the following statements about the complexity classes P and NP.

    - If P equals NP then the traveling salesperson problem has a polynomial time algorithm.
    - If the traveling salesperson problem has a polynomial time algorithm then the subset-sum problem has a polynomial time algorithm.
    - If the subset-sum problem has a polynomial time algorithm then P equals NP.

    How many of the above statements are true?

    1. 0 (none)
    2. 1
    3. 2
    4. 3 (all)

12. Consider lock variables a and b. They have a *test* method, which returns $True$ if the lock is available. That is in addition to the standard lock and unlock methods to acquire and release the lock. Which of the following fragments (used by every threads) can cause deadlock

A

```
while(a.test()):
   while(!b.test()):
      a.lock()
      b.lock()
-- critical section --
b.unlock()
a.unlock()
```

B

```
while(!a.test() || !b.test()):
    ;
a.lock()
b.lock()
-- critical section --
a.unlock()
b.unlock()
```

    1. A
    2. B
    3. both
    4. neither

13. In a multi-threaded process, the context for executing threads **does not** include:

    1. page table entries
    2. program counter
    3. registers
    4. stack pointer

14. In a virtual memory system, TLB:

    1. contains page table entries
    2. is translation lower bound
    3. contains tag lookup buffer
    4. is organized into pages

Name: _____ **Application number**: _____

PhD / MS(R) Admission Exam: Sample Duration: 2 hours

1.2 Numerical Answer

The answer to each of the following questions is a number. If the answer is an integer, write it in decimal without leading zeros. If it is a rational number, write it in the numerator/denominator form, where numerator and denominator are both integers written in decimal without leading zeroes, such that their GCD is 1. Correct answer gets 4 marks, while a wrong answer and no answer both get 0 marks.

1. The probability that a uniformly random IITD professor understands general relativity is 3/5. The probability that a uniformly random IITD professor understands quantum physics is 7/10. Under these assumptions, what is the minimum possible value of the probability that a uniformly random IITD professor understands both general relativity and quantum physics?

2. Seven cards have one integer written on each of their two sides. They are placed on a table in front of you. You can see the numbers $0, 1, 3, 5, 6, 7, 11$ written on the cards, but you have no idea of the numbers written on their sides facing the table. Your friend says, "For each of these seven cards, if the number on any one side is divisible by 2, then the number on the other side is not divisible by 3". You are allowed to specify a subset S of cards to flip in order to verify whether your friend's claim is true. What is the size of the smallest such S that is guaranteed to do the job?

3. A (simple and undirected) graph G is called a *pseudoforest* if every connected component of G contains at most as many edges as the number of vertices in that component. What is the maximum possible number of edges in a pseudoforest on a set of 2024 vertices?

4. S is a set with 8 elements. Let n denote the number of pairs of sets (A, B) such that $\emptyset \subseteq A \subseteq B \subseteq S$. What is $\log_9 n$?

5. You were a judge of a season of the reality show "Mastercook India" which started out with n contestants. In each episode of the show, all the surviving contestants cooked a dish for you. You tasted all dishes and eliminated the one contestant with the worst dish. After $n-1$ of the n contestants were eliminated, the unique surviving contestant was declared as the winner. You are unable to remember n but you do remember that the number of dishes you tasted is between 220 and 235. What is n?

6. Among the applicants to the Ph.D. program at IIT Delhi, 54 applicants know Java, 39 know Python and 43 know C++. Of these, 15 know both Java and Python, 17 know both Python and C++, 23 know both Java and C++, and 11 know all three languages. If there are 100 total applicants, how many know none of these three languages?

7. A hypothetical computer has main memory that can accommodate 4 pages. The operating system uses the "Least Recently Used" policy for page replacement. Suppose that the main memory contains pages $0, 1, 2, 3$ and gets requests to pages $4, 4, 5, 4, 5, 6, 4, 5, 6, 7, 4, 5, 6, 7, 8, 4, 5, 6, 7, 8, 9$. What is the number of page faults resulting from this sequence of requests?

8. What is the maximum possible number of grandparents in a binary tree with 2024 nodes?

9. There are 3 coins. Coin A has heads on both sides. For coin B heads comes up 75 percent of the time and coin C is a fair coin. A coin is selected uniformly at random and flipped. Given that the coin that was flipped comes up head, what is the probability that it was the fair coin?

10. The maximum latency of the following code is _____

```
0: R0 <- A[0]
1: R1 <- A[1]
2: R0 += R1
3: R2 <- A[2]
```

```
4: R0 += R2
5: R3 <- A[3]
6: R0 += R3
7: A[0] <- R0
```

given that all memory instructions have a fixed latency of 5 clocks and compute instructions have a fixed latency of 1 clock. The architecture is able to issue up to 3 instructions per clock and supports out of order execution (with a large look ahead window).

2 Programming Problems (48 marks)

This section consists of programming problems. The permissible languages for writing programs are `C++` and `python`. The programs are to be written on the moodle VPL interface. The detailed instructions will be given before the exam. For now, just try writing programs for the following tasks on your computers and run it on the provided test cases. The marks for each problem are equally divided among the test cases. Completely correct answer for each test case gets full marks for that problem.

2.1 Full code evaluation on VPL

1. **Sweet Distribution Dilemma – 18 marks**

   You took `N` packets of sweets to distribute among your `M` friends (`N` $\geq$ `M`, always handy to keep extra!) on your birthday. However, the person who packed these sweets made a mistake, putting a variable number of sweets in each packet. Since it is too late to redistribute the sweets within the packets, you decide to distribute the packets in an almost fair manner, ensuring:

   - Each friend gets exactly one packet.
   - The absolute difference between the number of sweets received by any pair of friends is minimized.

   Formally, the unfairness of a distribution is the difference between the maximum and minimum number of sweets any of your friends received. You want to distribute packets so that unfairness is minimized.

   Input format: The first line of input contains two space-separated nonnegative integers: `N`, the number of packets of sweets you have, and `M`, the total number of your friends. The second line of input contains `N` space-separated positive integers, where the i'th number represents the number of sweets in the i'th packet.

   Output format: Print a single integer, which is the unfairness of an unfairness-minimizing distribution of packets to friends, subject to the constraint that every friend receives exactly one packet.

2. **Efficient Parcel Delivery – 18 marks**

   You run a parcel delivery business and have `n` trucks to do so. Given a client's order to deliver `m` identical packages, you find that your i' th truck can hold up to c_i packages. You wish to determine the fewest number of trucks that you must send to accommodate a total of `m` parcels.

   Input format: The first line of input contains two space-separated nonnegative integers: `m`, the number of parcels to be delivered and `n`, the number of trucks. The second line of input contains `n` space-separated positive integers – the capacities of the `n` trucks.

   Output format: Print a single number, which is the minimum number of trucks needed to fit all the `m` parcels.

2.2 Complete given code – 12 marks

In this task, you are asked to find the sum of two numbers. The numbers in this system are represented as an arbitrarily long list of nodes, each containing a single digit. Do this by replacing the dashes with appropriate code in the function addList. You are also provided with code to test – It reads each list as space separated digits followed by the end of line. It then prints the computed sum, also as space separated digits. For example, if the following two lines are input:
2 3 4
6 7 9

The answer printed is:
0 9 1 3

C++ Version

```
#include <string>
#include <iostream>
#include <sstream>

typedef char digit;
const int base = 10;

struct Node {
   // Used to create a list of digits
   struct Node *nextNode;
   digit value;
   Node(digit v, Node *n=NULL) { value = (digit)v; nextNode=n;}

   void print() { std::cout << (int)value << " "; }

   void printAll() {
      print();
      if(nextNode != NULL) nextNode->printAll();
   }

};

digit sum1(digit a, digit b, digit c)
{ // Returns the digit sum on adding digits a, b, and carry c
   return (digit)(((int)a+(int)b+(int)c)%base);
}

digit carry1(digit a, digit b, digit c)
{ // Returns the carry digit on adding digits a, b, and carry c
   return (digit)(((int)a+(int)b+(int)c)/base);
}

Node *addList(Node *n1, Node *n2) // Complete this function
{ // List n1 and n2 are interpreted as 0-padded numbers, starting with the most
  // significant digit in the first node. Both numbers have the same number of digits.
  // This function produces a new list with the sum of input numbers.
  // The first node of the result list contains the carry digit (either 0 or 1).

   if(n1 == NULL) return NULL; // n2 would also be NULL
   Node *sum = addList(_______________);
   digit carry = (sum == NULL)? 0:sum->value;
   digit s1 = sum1(n1->value, n2->value, carry);
   digit c1 = carry1(n1->value, n2->value, carry);
   if(________)
      sum = new Node(s1); // Allocates Node and sets its value to s1
   else                   // (if you not familiar with C++)
```

```
      ____________________
   return(__________________);
}

struct Num {
   Node *first;
   Num(Node *f = NULL) { first = f; }

   Num(std::istream &cin) {
      first = NULL;
      std::string digits;
      getline(cin, digits);
      std::istringstream dstream(digits);

      int d;
      if(dstream >> d) {
         first = new Node(d);
         Node *p = first;
         while(dstream >> d) {
            p->nextNode = new Node(d%base);
            p = p->nextNode;
         }
      }
   }

   void print() { if(first != NULL) first->printAll(); std::cout << std::endl; }
   Num addNum(const Num &n) { return Num(addList(first, n.first)); }
};

int main(int argc, char *argv[])
{
   Num num1(std::cin);
   Num num2(std::cin);
   num1.addNum(num2).print();
}
```

Python Version

```
class digit(int):
   def __new__(cls, v):
      return super(digit, cls).__new__(cls, int(v)%digit.base)
   base=10

class Node:
   # Used to create a list of digits
   def __init__(self, v, n=None):
      self.value = digit(v) # The digit at this node
      self.nextNode=n # Reference to the next digit's node

   def printAll(self):
```

```
      print(self.value, end=" ")
      if(self.nextNode != None): self.nextNode.printAll()

def sum1(a, b, c):
   # Returns the digit sum on adding digits a, b, and carry c
   return digit((int(a)+int(b)+int(c))%digit.base)

def carry1(a, b, c):
   # Returns the carry digit on adding digits a, b, and carry c
   return digit((int(a)+int(b)+int(c))/digit.base)

def addList(n1, n2): # Complete this function
   # List n1 and n2 are interpreted as 0-padded numbers, starting with the most
   # significant digit in the first node. Both numbers have the same number of digits.
   # This function produces a new list with the sum of input numbers.
   # The first node of the result list contains the carry digit (either 0 or 1).
   if(n1 == None): return None # n2 would also be None
   sum = addList(_______________)
   carry = 0 if (sum == None) else sum.value
   s1 = sum1(n1.value, n2.value, carry)
   c1 = carry1(n1.value, n2.value, carry)
   if(____________):
       sum = Node(s1)
   else:
      ____________
   return(_______)

class Num:
   # If node is not provided, read from stdin
   def __init__(self, f = None):
      self.first = f
      if(f == None):
         digits = [digit(i) for i in input().split()]

         for d in digits:
            if self.first == None:
               p = self.first = Node(d)
            else:
               p.nextNode = Node(d)
               p = p.nextNode

   def print(self):
     if(self.first != None):
        self.first.printAll()
        print()

   def addNum(self, n): return Num(addList(self.first, n.first))

num1 = Num()
num2 = Num()
```

Name: _____ **Application number**: _____

PhD / MS(R) Admission Exam: Sample Duration: 2 hours

```
num1.addNum(num2).print()
```

Name: _____ **Application number**: _____

PhD / MS(R) Admission Exam: Sample Duration: 2 hours

3 Subjective Questions

1. Draw a Cartesian coordinate system with the (horizontal) x -axis and the (vertical) y -axis extending from -2 to 2. Sketch the function $y = \log_2 |x^2 - 1|$. Label each point of intersection of this curve with the axes with its coordinates.

2. We have a new kind of wantem computer, which can compare up to any n pairs of integers in an array of size n in one time step. Provide a $\theta(\log n)$-time algorithm to sort an array of integers. Derive its time complexity.

3. Draw the tallest AVL tree (i.e., one having the greatest possible height) with 6 nodes.

4. We, the CSE department at IITD, are thinking of switching to the following algorithm for hiring research students: interview the n applicants in a uniformly random order, and hire every applicant who is better than all the previously interviewed applicants. Assume that no two applicants are ever found to be equally good. Show your steps.
   (a) What is the probability of us hiring the i'th candidate to be interviewed?
   (b) Let $H(n)$ denote the expected number of candidates out of n that get hired out of this process. Please help us by specifying the asymptotic behavior of $H(n)$ using the $\Theta(\cdot)$ notation.