Introduction to Computer Science (COL100) **Major** **Sept 12, 2020**

Note: Maximum marks : 60 . All notations standard (as done in class). Duration 3 hours.
Instructions:

1. Log in to `col100.iitd.ac.in`. Create a directory `major` in your home folder (note the case sensitivity).

2. Solution to each question should be saved in a separated file such as q1.soln, q2.soln (again note the case sensitivity) and so on in the created directory.

3. All analyses should be performed inline.

4. Declarations of honesty should be prepared for each question and should be saved in the question's solution file.

5. **Penalty:** A 10% of maximum marks for up to the first hour after the duration, 20% for the second hour, and 50% for the rest of the day. 100% penalty would be applied thereafter.

---

1. (10 Marks) Consider two matrices A and B with the same dimension of $n \times n$. Write a program to multiply A and B and store the result in a matrix C. Assume the elements of A, B and C are given as 1-D arrays where a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is stored as an array $Arr = [a, c, b, d]$.

2. (10 Marks) Define classes for `Point`, `Line`, and `Circle` to implement Ruler-Compass algebra. Assume the existence of the following methods (i.e. you do not have to define them):

   - Point(int x, int y), Line(point a, point b), Circle(point a, int radius): creates a point, line, and circle objects, respectively.
   - intersectTwoLines(Line a, Line l): returns a point or throws an exception
   - intersectTwoCircles (Circle c1, Circle c2): returns either a pair of points *lexicographically* (i.e. sorted first according to x-coordinate and then according to y-coordinate) or throws an exception.
   - intersectLineCircle(Line l, Circle c): returns either a pair of points ordered *lexicographically* on the line or throws an exception.
   - getLengthWithCompass(Point a, Point b): returns the length between two points

   (a) Place the method declarations of given functions in suitable classes clearly showing method return types, if any.

   (b) Implement the following methods:(i) drawing the perpendicular from a point on a given line, (ii) square root of an integer, and (iii) gcd of two non-negative integers.

3. (10 Marks) The following pattern is called the *Pascal's triangle*. The numbers at the edge of the triangle are all 1, and each number inside the triangle is the sum of the two numbers above it. For example, the first row is described as $pascal(0,0) = 1$, the second row is descibed by $pascal(1,0) = 1$ and $pascal(1,1) = 1$, the third row is descibed by $pascal(2,0) = 1$, $pascal(2,1) = 2$, $pascal(2,2) = 1$, and so on.

$$
\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1
\end{array}
$$

Develop an algorithm, as efficient as possible in both time and space, that computes an element of Pascal's triangle $pascal(p, q)$ given p and q.

4. (10 Marks)

   (a) Provide one-line answers to the following with reasons:

      i. Are floating point numbers finite?

      ii. Are floating point numbers dense?

      iii. Are floating point numbers commutative ($i.e.$, $a \oplus b = b \oplus a$) with respect to addition and multiplication?

      iv. Are floating point numbers associative ($i.e.$, $a \oplus (b \oplus c) = (a \oplus b) \oplus c$) with respect to addition and multiplication?

      v. Do you foresee any problems in doing Calculus with floating point numbers?

   (b) Solve the system

$$
\begin{array}{rcrcl}
2x & - & 4y & = & 1 \\
-2.998x & + & 6.001y & = & 2
\end{array}
$$

   using any method you know. Compare the solution with the solution to the system obtained by changing the last equation to $-2.998x + 6y = 2$. Is this problem stable? Please provide reasons for your answer.

5. (20 Marks) In the second online test some of you implemented the junior school algorithm for multiplying two large integers in array representation. If both the input integers are of size n the time complexity of the algorithm is clearly $O(n^2)$. Here is an improved algorithm based on a divide and conquer strategy with a time complexity of $O(n^{\log 3}) \approx O(n^{1.59})$:

Let x and y be two integers with n and m digits respectively. Assume $n \geq m$. Partition x and y into four integers a, b, c and d such that

$$x = (a * 10^k + b)$$
$$y = (c * 10^k + d)$$

where $k = n$ div 2. Now,

$$z = x * y = (a * 10^k + b) * (c * 10^k + d) = a * c * 10^{2k} + (a * d + b * c) * 10^k + b * d$$

The above can be computed, recursively, using three multiplications as follows:

$$u = (a + b) * (c + d); v = a * c; w = b * d;$$
$$z = v * 10^{2k} + (u - v - w) * 10^k + w$$

Note that if x has n_1 digits and y has n_2 digits, then the time complexity of both addition and subtraction is $O(\max n_1, n_2)$.

(a) Why is it necessary to reduce to three multiplications instead of four?

(b) Assuming that both integers are of size $n = 2^k$ for some k (have n digits), give a recurrence that describes the time complexity of the algorithm. Solve the recurrence or prove by induction that the solution is $O(n^{\log 3})$.

(c) Assume that x and y are given either as arrays or lists of size m and n. The least significant digits of x and y, in the case of arrays are stored in $x[0]$ and $y[0]$ respectively, and in the case of lists are stored at $hd(x)$ and $hd(y)$ respectively. Write a Python function for multiplying them in their array representation or an ML function in their list representation.