INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Monsoon Semester, 2017
Algorithms (ISC211)

Assign. III
August 23, 2017

Due: **29.08.17**

Instructor: Dr. Pawan Kumar

Maximum Marks: 100

Instructions: Write answers on your own. Don't write unnecessary text. Some solutions to this assignment will be shown during the tutorial on 30/08/17 at 3:30-5:00PM in room 105.

1. You are using bucket sort (see Cormen book) to sort n numbers between 0 and 1. You have sorted the data into buckets. However, later you come to know that the data was not uniformly distributed in the given range. How will this affect your time complexity? Memory complexity? How will you minimize the time complexity assuming you still have to use the buckets (write the pseudo code). What is your new time complexity and memory complexity? [20]

2. Read topological sort from Cormen's Algorithm book. Write an algorithm, and derive the time complexity for this algorithm. How/When topological sort helps in Dynamic programming (DP)? [10]

3. For the memoized bottom-up DP algorithm of the rod cutting example shown in the class:
   1. Derive the recurrence relation for time $T(n)$.
   2. Determine the time complexity in n, where n is the length of the rod.
   3. Find a problem different from rod-cutting for which same algorithm (with a slight change that it could be minimization or maximization of something) will be applicable. [20]

4. There are n fields located in a circle around a hydroelectric power plant. Each field has the space to construct a power distribution station of varying capacity. The i^{th} power distribution station, if constructed, will draw p_i Watts of power from the power plant. However, due to environmental regulations, no two adjacent fields can have power distribution stations on both of them. Construct the power distribution stations in such a way as to maximize the total power drawn from the hydroelectric power station. [25]

5. Come up with the most optimal algorithm to find the number of permutations of numbers from $1 \ldots N$ satisfying the constraints as represented by a boolean array A of size N in the following manner:
   1. If $A[i] == 1$, then i should lie to the left of $i+1$ in the permutation.
   2. If $A[i] == 0$, then i should lie to the right of $i+1$ in the permutation. [25]

Student's name:

End of Assignment