INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

B.TECH. MID SEMESTER EXAMINATION, AUTUMN 2012-13
ALGORITHMS II (CS31005)

Full marks: 60 **Time: 2 hours**

1. Answer Question 1, Question 2, and any two from the rest.
2. All parts of a particular question should be answered together.
3. Credits will be given for neat and to-the-point answering. Unnecessary or redundant words are liable to negative marking.

1. Given a set $P = \{p_1, p_2, \ldots, p_n\}$ containing n points on the 2D plane, such that all the $\binom{n}{2}$ point-pair distances in P are distinct. Suggest an algorithm to prepare a 1D array A of size n in which $A[i] = j$ implies that out of all points in P, p_j is nearest to p_i.

   Explain its time complexity. (15+5)

2. Given n *types* of 3D boxes, where the ith box has dimensions x_i, y_i, z_i (all real numbers). Suggest an algorithm to create a stack of boxes which is as tall as possible, such that the dimensions of the 2D base of each box are strictly larger than the corresponding dimensions of the 2D base of its upper box (Condition C). Any of the six faces of a particular box may work as its base and a particular type of box may be used more than once, provided Condition C is not violated.

   Explain the time complexity of your algorithm. (15+5)

   An example: Given 2 types of boxes: $B_1(2 \times 3 \times 4)$ and $B_2(3 \times 4 \times 5)$. We can stack B_2 (base 4×5), then B_2 again (base 3×4), and then B_1 (base 2×3), which yields the tallest stack of height $3 + 5 + 4 = 12$.

3. In a dinner party, there are 10 couples. Each couple knows exactly 5 other couples. Dinner will be served in 10 tables, each table having seats for exactly two persons. Each man wishes to dine with some lady (apart from his wife) whom he knows, and each lady with some man (apart from her husband) whom she knows. Suggest an algorithm to make the dinner a grand success! (10)

4. Describe an efficient algorithm to compute a^4 where a is an n-bit binary integer. For simplicity, assume that n is in the power of 2. Give a recurrence relation for the running time of the algorithm, and solve the recurrence. (10)

5. Given a set S of n distinct integers. Propose a randomized algorithm to quick-sort S in an expected $O(n \log n)$ time. Justify the expected time complexity. (10)