COV 886 Special Module in Algorithms
Assignment Sheet

Last updated March 17, 2018

1. Extend the RIC for closest pair in two dimensions to closest pair in three dimensional space. Provide complete analysis along with details of any geometric properties and data structures used.

2. Consider the trivial algorithm for selecting the minimum element among n elements. We scan the elements given in an array $A[]$ and keep track of the minimum elements among the elements scanned so far - call it the *temporary minimum*. At the i-th step, we either upadate the minimum or we retain it. If the elements are presented to us in a random order, what is the expected number of times that we update the temporary minimum.

3. Complete the analysis of the incremental Quicksort described in class by filling in complete details.

4. How would you choose a random subset of r elements from a given set of n elements ($r < n$) ?
Prove a bound on the maximum size of the intervals similar to binomial sampling.

5. *Convex hull* Consider the following incremental algorithm for constructing a two dimensional (planar) convex hull of a given set S of n points. Let S_i denote set of the first i points (randomly chosen) and $CH(S_i)$ is the convex hull of S_i. For the remaining points $S - S_i$, we maintain an intersection point on the boundary with a point inside the convex hull - say, O. (It is very easy to find such a point O). For a point p_j, let us denote the (latest) intersection points by q_j. Note that if point is inside the hull, then q_j are not defined (neither necessary).

   When the next point p_{i+1} is added, we start a walk in the clockwise/counter-clockwise directions from q_{i+1}. We can determine the two points of tangency easily this way and update the convex hull to $CH(S_{i+1})$. We now need to update the q_js (or to determine that that they are undefined).
   (i) How do you update the new q_js for the uninserted points ?
   Hint: It must intersect with one of the new edges.
   (ii) What is the expected cost of the update step and based on this analyse the expected running time for the algorithm.

6. *Smallest enclosing ball* Given a set of n points, S, we would like to construct the smallest ball $\mathcal{B}$ containing S (points can be on the boundary). Design a linear time algorithm based on randomized incremental construction. You can assume that no more than three points lie on any circle.

7. Consider the recurrence for the 2 dimnsional linear programming based on RIC.

$$T(n,2) = \begin{cases} O(n) \text{ if } d = 1, \\ O(1) \text{ if } n = 1, \\ T(n-1,2) + \frac{2}{n}(O(n) + T(n-1,1)) + O(1) \text{ otherwise} \end{cases}$$

   Show that $T(n,2) = O(n)$.
   For the general recurrence in d dimension show that the solution of the recurrence

$$T(n,d) = O\left(\sum_{i=1}^{d} \frac{i^2}{i!} d!n\right)$$