6.893 Sub-linear Algorithms	December 1, 2014
Homework 3	
Lecturer: Ronitt Rubinfeld	*Due Date: December 15, 2014*

Turn in your solution to each *problem on a separate sheet of paper, with your name on each one.*

MORE QUESTIONS COMING SOON!

1. (Property testing of the clusterability of a set of points.) Given a set X of points in any metric space. Assume that one can compute the distance between any pair of points in one step. Say that X is (k, b)-diameter clusterable if X can be partitioned into k subsets (clusters) such that the maximum distance between any pair of points in a cluster is b. Say that X is ϵ-far from (k, b)-diameter clusterable if at least $\epsilon|X|$ points must be deleted from X in order to make it (k, b)-diameter clusterable.

   Show how to distinguish the case when X is (k, b)-diameter clusterable from the case when X is ϵ-far from $(k, 2b)$-diameter clusterable. Your algorithm should run in time polynomial in $k, 1/\epsilon$. It is possible to get an algorithm which runs in time $O((k^2 \log k)/\epsilon)$.

2. A *vertex cover* V' of a set of edges E' is a set of nodes such that every edge of E' is adjacent to one of the nodes in V'.

   For graph $G = (V, E)$, let the *transitive closure graph* $TC(G)$ be the graph $G^{(tc)}(V, E^{(tc)})$ where $(u, v) \in E^{(tc)}$ if there is a directed path from u to v in G.

   Let $f : V \to \{0, 1\}$ be a labeling of the vertices of a known directed acyclic graph G by 0 and 1. For any pair of nodes x and y, we say that $x \leq_G y$ if there is a path from x to y in G. We say that f is *monotone* if for all $x \leq_G y$, $f(x) \leq f(y)$. The *minimum distance of f to monotone* is the minimum number of nodes that must be relabeled in order to turn f into a monotone function.

   Let E' be the set of violating edges in $TC(G)$ according to f. Show that the minimum distance of f to monotone is equal to the minimum size of a vertex cover of E'.

3. In the Run Length Encoding (RLE) compression scheme, the data is encoded as follows: each run, or a sequence of consecutive occurrences of the same character, is stored as a pair containing the character in the first location and the length of the run in the second location.[1] For example, the string 11111101000 would be stored as $(1, 6)(0, 1)(1, 1)(0, 3)$. The *cost of the run-length encoding*, denoted by $C(w)$, is the sum over all runs of $\log(\ell + 1) + \log|\Sigma|$.

   Assume that the alphabet characters are all in the set $\{0, 1\}$, i.e., that the alphabet Σ is of size 2.

   (a) Give an algorithm that, given a parameter ϵ, outputs an ϵn-additive estimate[2] to $C(w)$ with high probability and makes $\mathrm{poly}(1/\epsilon, \log n)$ queries.

[1] Run-length encoding is used to compress black and white images, faxes, and other simple graphic images, such as icons and line drawings, which usually contain many long runs.

[2] For a function f, algorithm A outputs an ϵn-additive estimate if on any input x, $f(x) - \epsilon n \leq A(x) \leq f(x) + \epsilon n$.

(b) Show that there is a distribution on inputs such any that any deterministic approximation algorithm for $C(w)$ making an expected number of queries that is $o(\frac{n}{A^2 \log n})$ must fail to output an A-multiplicative approximation with probability at least $1/3$. (Here the expectation in the number of queries is over the choice of an input from the distribution). (It's also ok to give a lower bound for deterministic algorithms by showing that for each algorithm there is an input that causes it to fail).