CS 466/666, Spring 2009
Timothy Chan

Assignment 3 (due July 2 Wednesday 5pm)

Please read `http://www.student.cs.uwaterloo.ca/~cs466/policies.html` first for general instructions.

1. [*14 marks*] Given a set S of n points (the "scene") and a set P of m points (the "pattern") in two dimensions ($m \leq n$), a basic question in pattern recognition is to see whether the pattern "occurs" in the scene. Due to the possibility of noise in the data, an exact match is not required and we are satisfied if, say, 90% of the pattern points are matched.

   Here is one precise formulation of the problem that you will study: determine whether there exists a translation of P so that at least $0.9m$ of the translated points in P are in S.

   (You may assume that testing whether a point is in S can be done in constant time by hashing.)

   (a) [*4 marks*] Describe a deterministic algorithm that solves this problem in $O(mn^2)$ time. *Hint*: try all pairs $p \in P$ and $q \in S$ and consider the translation that takes p to q...

   (b) [*10 marks*] Give a Monte-Carlo algorithm that runs in $O(n^2)$ time with error probability at most 0.01. [Hint: choose $p \in P$ at random...]

   Bonus (3 marks): Give a Monte-Carlo algorithm that runs in $O(n)$ time, with error probability at most 0.01, under the assumption that no two pairs of points in S have the same exact distance.

2. [*16 marks*]

   (a) [*3 marks*] Given a set S of n numbers and a subset $R = \{x_1, \ldots, x_r\}$ where each element x_i is a random, independently chosen element from S, compute the probability that $\min(R)$ has rank more than k (i.e., there are at least k elements with value smaller than $\min(R)$).

   (b) [*3 marks*] Show that $\min(R)$ has rank at most $c(n/r)\log n$ with probability at least $1 - 1/n$, for some constant c. [Note: the inequality $1 - x \leq e^{-x}$ may be useful...]

   (c) [*10 marks*] You are given an undirected graph $G = (V, E)$ stored in the standard adjacency-lists representation. Assume that each vertex has degree at most a constant, say 3. Assume each vertex v holds a real-valued weight $w(v)$. The problem is to find a *local minimum* u, i.e., a vertex u satisfying the property that $w(u) \leq w(v)$ for all vertices v adjacent to u. (There could be more than one local minimum.)

   Surprisingly, with randomization it is possible to solve this problem without having to examine all n vertices of the graph! Give a Las Vegas algorithm that finds a local minimum in $O(\sqrt{n \log n})$ time with probability $1 - 1/n$.

   [Hint: use random sampling. As you would expect, part (b) should come in handy...]

3. [*20 marks*] Consider the following problem: given a binary string $u = a_1 a_2 \dots a_n \in \{0,1\}^*$, decide whether u contains 00 as a substring (i.e., whether u contains two consecutive 0's). Obviously, there is an $O(n)$-time algorithm. We will investigate the precise number of bits that need to be examined (i.e., number of a_i's that need to be evaluated) for this problem.

   (a) [*10 marks*] Prove that every deterministic algorithm needs to evaluate at least $cn - o(n)$ bits in the worst case for some constant $c \geq 0.75$.

   [Note: A proof for a weaker constant, e.g., $c = 1/2$ or $c = 2/3$, will receive some partial marks. Hint: use an adversary argument. Fix $a_1 = a_5 = \cdots = 1$ and try to force all positions $a_2, a_3, a_4, a_6, a_7, a_8 \dots$ to be evaluated.]

   (b) [*10 marks*] Give a Las Vegas algorithm that only evaluates an expected $c'n + o(n)$ number of bits for any given input string for some constant $c' < 1$.

   [Note: A correct proof for any constant strictly smaller than 1 will receive full marks. Hint: for one possible approach, consider the following observations: (i) if $a_i = a_{i+2} = 1$, then we do not need to evaluate a_{i+1} (why?); (ii) if $a_1 a_2 a_3 a_4 a_5$ does not contain 00, then $a_1 = a_3 = 1$ or $a_2 = a_4 = 1$ or $a_3 = a_5 = 1$ (why?).]