CS 466/666, Spring 2006
Timothy Chan

**Assignment 4 (due July 13 Thursday noon)**

1. [*12 marks*] Consider a variant of the random walk scenario: An ant is placed on an interval $[0, n]$. At position $i$, the ant moves to position $i-1$ with probability 1/2 and position $i+2$ with probability 1/2. The exceptional cases are when $i = n$ and $i = n-1$: here, the ant moves to position $i-1$ with probability 1. At position $i = 0$, the ant stops.

   Prove that the expected number of steps before the ant stops starting at any position $i \neq 0$ is at least exponential: more precisely, $\Omega(\phi^n)$, where $\phi = (1+\sqrt{5})/2$.

   [Hint: define $t_i$ as in class and $d_i = t_i - t_{i-1}$. As you may guess, Fibonacci numbers come up somehow (you may use the fact that the $k$-th Fibonacci number has growth rate $F_k = \Theta(\phi^k)$). You do not need to compute $t_i$ exactly; just lower-bound the value of $t_i$.]

2. [*16 marks*] Given a set $P$ of $n$ points in 2D and an integer $k$, we wish to select a subset of $k$ points that are as "widely scattered" as possible. One way to formulate the problem is as follows: find a subset $Q$, such that $|Q| = k$, maximizing $\delta(Q)$, where $\delta(Q) = \min_{q,q' \in Q,\ q \neq q'} d(q, q')$ denotes the distance of the closest pair in $Q$. (The notation $d(q, q')$ denotes the Euclidean distance between two points $q$ and $q'$.)

   Consider the following greedy algorithm:

   0. $Q = \emptyset$
   1. for $i = 1, \ldots, k$ do {
   2.     find a point $q \in P$ that maximizes $\delta(Q \cup \{q\})$
   3.     insert $q$ to $Q$
   4. }
   5. return $Q$

   (a) [*3 marks*] Show that this algorithm can be implemented in $O(nk^2)$ time. [Hint: how efficiently can you compute $\delta(Q \cup \{q\})$, knowing $\delta(Q)$?]

   (b) [*2 marks*] Let $q_i$ denote the $i$-th point inserted to $Q$ by the greedy algorithm. Show that for every point $p \in P$, $\delta(\{q_1, \ldots, q_i\}) \geq \delta(\{q_1, \ldots, q_{i-1}, p\})$.

   (c) [*5 marks*] Let $r = \delta(Q)$ for the subset $Q$ returned by the algorithm. Let $D_i$ denote the circular disk centered at $q_i$ of radius $r$. Prove that every point $p \in P$ lies in the union of the $k-1$ disks $D_1 \cup \cdots \cup D_{k-1}$.
   [Hint: prove by contradiction; say $r = d(q_i, q_j)$ for some $i > j$ and apply part (b)...]

   (d) [*5 marks*] Show that the greedy algorithm achieves approximation factor at least 1/2.
   [Hint: recall the pigeonhole principle...]

3. [*10 marks*] Let's reexamine the problem in Question 2, this time, in nongeometric settings: given a set $P$ of $n$ elements, an integer $k$, and a distance matrix $d(\cdot,\cdot)$, find a subset $Q$, such that $|Q| = k$, maximizing $\min_{q,q' \in Q,\, q \neq q'} d(q,q')$. We are now concerned with lower bounds:

   (a) [*5 marks*] Prove that if P $\neq$ NP, then there is no polynomial-time approximation algorithm for the problem in general that achieves any approximation factor $\rho$ for any constant value $\rho$.

   (b) [*5 marks*] Now, consider the restricted setting where $d(\cdot,\cdot)$ is a metric (satisfying symmetry and triangle inequality). Prove that if P $\neq$ NP, then there is no polynomial-time approximation algorithm for the problem in the metric case that achieves approximation factor strictly better than 1/2.

   [Hint: you may assume that the decision problems Clique or Independent-Set for graphs are NP-complete (e.g., see [CLRS]).]

4. [*12 marks*] Given a set $S$ of $n$ unit squares in 2D, we want to assign a color to each square so that no two intersecting squares have the same color, while minimizing the number of colors used.

   Design an approximation algorithm for the problem with approximation factor at most 4, by using a grid approach. (As in class, consider the centers of the squares and assign centers to grid cells.)