1. [42 points]

(a) What is the solution to the recursion formula
\[ T(n) = n^2 + 2T\left(\frac{n}{2}\right) \]
(b) Is the following family of functions a universal family: \( H = \{ h_1, \ldots, h_n \} \) where \( h_i(x) = x \mod 7 \), and \( i \) is a value in the set \( \{0, 1, 7, 8, 14, 15, 21, 22, \ldots, 7k, 7k + 1\} \) for \( 0 \leq k \leq 1000 \)? Assume \( U = [1 : \ldots : 2^{32}] \).
(c) Assume a universe \( U = [1 : \ldots : 2^{32}] \), and a hash table \( T[1 : \ldots : m] \), and \( m = 256 \). Is there a hash function \( h(x) \) such that \( 1 \leq h(x) \leq m \) for every \( x \in U \), and in addition, for every set \( K \) of \( m/2 \) keys from \( U \), the function \( h \) is a perfect hash function? That is, \( h(x) \neq h(y) \) for every pair \( x, y \in K \).
(d) In hw1, you were asked to suggest an \( O(n) \) time algorithm for computing the upper envelope \( U(S) \) from a set \( S \) of \( n \) lines. It was given in the question that the lines in \( S \) are sorted according to their slopes. Explain briefly which properties in your algorithm are not true if the lines in \( S \) are not sorted. In particular, would your algorithm work in this case?

2. [18 points]

To handle the semi-dynamic Voronoi Diagram for a set \( S \) of \( n \) points in the plane, the following algorithm is proposed: Let \( n_0 \) denote a *xed large number indicating the maximum size of input we might need to handle. So \( n \leq n_0 \). We maintain two disjoint sets \( S_1 \) and \( S_2 \) such that \( S = S_1 \cup S_2 \), and \( jS_1 \leq p_{n_0} \). We compute only the Voronoi Diagram \( \text{VD}(S_2) \) of \( S_2 \). We maintain the following operations:
- \( \text{query}(q) \). This operation finds the nearest neighbor to a query point \( q \).
- \( \text{insert}(x) \). This operation inserts a new point to \( S \). To support this operation, we use \( \text{VD}(S_2) \) to find in \( O(\log n) \) the nearest neighbor to \( q \) in \( S_2 \). In addition, we check the distance from \( q \) to every point in \( S_1 \). The nearest distance is reported.
ation, we first add $x$ to $S_1$. Then if $|S_1| \geq n$, we add to $S_2$ all points of $S_1$, compute $VD(S_2)$ (which takes $O(n \log n)$) and empty $S_1$.

What is the amortized running time of each operation?

3. [18 points] Consider a SkipList constructed on the keys $x_1, x_2, \ldots, x_n$, where $x_1 < x_2 < \ldots < x_n$. Assume that we perform $\text{search}(x_n)$. What is the probability that the search path passes through at least $t$ keys all at the lowest level of the SkipList? Here $t$ is some given parameter.

4. [25 points] Answer one of the two questions below, but not both.

(a) While parachuting, the wind has carried you to the desert, and you do not know where you have landed. You have no map. You have a GPS device, but all it can report is your current position. That is, it does not have any map built in.

Since the desert is a dangerous place, you wish to walk in the shortest way to a road. It does not matter to which road you will walk once you arrive at a road, you could be rescued. Suggest a path that you would follow, such that the total length you walk along the path until you reach a road is $O(d)$. Here $d$ is the distance from your landing point $q$ to the nearest point on the nearest road. Of course, you do not know $d$ in advance, and you do not know which direction leads to the nearest road. Prove your answer.

(b) Consider two sets of points $X = \{x_1, \ldots, x_m\}$ and $B = \{b_1, \ldots, b_l\}$. Each is a real number. We call the points in $X$ traffic lights and the points in $B$ blockages. We say that a point $q$ sees a traffic light $x$ if $q < x$ and in addition, the interval $(q; x)$ does not contain any blockage $b \in B$. Suggest a data structure that can support the following operations, each in $O(\log n)$ where $n = jXj + jBj$.

- $\text{InsertT}(x)$ insert a new traffic light at the point $x$.
- $\text{InsertB}(b)$ insert a new blockage at the point $b$.
- $\text{Count}(q)$ Report how many traffic lights of $X$ the point $q$ sees.