CSL 356, Problem Sheet 4

1. Show that for any collection of hash function H. there exists x, y such that

$$\sum_{h \in H} \delta_h(x, y) \geq |H|(\frac{1}{m} - \frac{1}{n})$$

where n and m are the sizes of universe and table respectively.
Remarks: This justifies the definition of universal hash function.

2. Assume that the size of the table T is a prime m. Partition a key x into $r+1$ parts $x =< x_0, x_1 \dots x_r >$ where $x_i < m$. Let $a =< a_0, a_1 \dots a_r >$ be a sequence where $a_i \in \{0, 1, \dots m-1\}$. We define a hash function $h_a(x) = \sum_i a_i x_i \text{mod} m$. Clearly there are m^{r+1} distinct hash functions. Prove that $\cup_a h_a$ forms a universal class of hash functions.

3. A collection of hash function H is called *strongly* universal if for all keys x, y and any $i, j \in [0..m-1]$

$$\Pr_{h \in H}(h(x) = i \wedge h(y) = j) \leq \frac{c}{m^2}$$

How does this differ from the earlier definition (in lecture) ? Can you give an example of a strongly universal family ?

4. Prove that for any (non-zero) vector over $\{0, 1\}$ of length n when multiplied by a random (0,1) vector (dot-product), the probability that it is 0 (summation is mod 2) is $\leq 1/2$.

   Use this fact to verify if for matrices A, B, C ($n \times n$ with 0, 1 entries)

$$AB = C.$$

   Additions are mod 2 and your algorithm should run in $O(n^2)$ steps and be correct with probability $\geq 3/4$. (These kind of randomized algorithms are Monte Carlo).

5. Given a set of n horizontal line segments, design a data structure that reports all intersections with a query vertical segment. Hint: Use segment trees.

6. Analyse the performance of range trees for reporting orthogonal range queries for dimensions $d \geq 3$. In particular what are the preprocessing space and query time ?

7. If we allow for insertion and deletion of points, how does the performance of range trees get affected ? In particular what are the time bounds for orthogonal range query, insertion and deletion of points ? Discuss the data structure in details.

8. Design efficient algorithms to construct *union* and *intersection* of two convex hulls.

9. A point $p_1 \succeq p_2$, (p_1 dominates p_2) if all the coordinates of p_1 is greater than all the coordinates of p_2. A *maximal* point is one that is **not** dominated by any other point in a given set S. The DOMINANCE problem is to find out all the maximal points in a given set S.
(i) Design an $O(n \log n)$ algorithm for the two dimensional DOMINANCE problem. (ii) Design an $O(n \log n)$ algorithm for the three dimensional version of the DOMINANCE problem.