1. **Open Addressing**
   Let table size be $m = 15$ (with items indexed from 0...14).
   Use the hash function $h(x) = (x \mod 15)$ and linear hashing to hash the items 19, 6, 18, 34, 25, 34 in that order.
   Draw the resulting table.

2. **Universal Hashing**
   Recall the universal hash function family defined by
   \[ h_{a,b}(x) = \left( (ax + b) \mod p \right) \mod m \]
   where $a \in \mathbb{Z}_p^*$, $b \in \mathbb{Z}_p$ and $p$ is a prime with $p \geq U$. Let $p = 17$, $m = 5$. For all $x = 0, 1, \ldots, 16$ write the values for $h_{1,0}(x)$. Now write all the values for $h_{2,2}(x)$.
3. Divide and Conquer for closest pair

Let \( P = \{p_1, p_2, \ldots, p_n\} \) be \( n \) two-dimensional points and define

\[
\delta(P) = \min_{p, p' \in P : p \neq p'} d(p, p')
\]

to be the closest pair distance of \( P \).

Let \( X \) be a real value and split \( P \) on the line \( x = X \) so that

\[
P_L = \{p \in P : p.x \leq X\}, \quad P_R = \{p \in P : p.x > X\}.
\]

Suppose you are given the closest pair distance of the two sets:

\[
\delta_L = \delta(P_L) \quad \text{and} \quad \delta_R = \delta(P_R).
\]

Set \( \delta' = \min(\delta_L, \delta_R) \) and define the points contained by the \( \delta' \) strips to the left and right of the line \( x = X \) by

\[
S_L = \{p \in P_L : X - p.x \leq \delta'\}, \quad S_R = \{p \in P_R : p.x - X \leq \delta'\}
\]

(a) Prove that

\[
\delta(P) = \min(\delta_L, \delta_R, d(S_L, S_R))
\]

where \( d(p_1, p_2) = \min\{d(p_i, p_j), : p_1 \in P_1, p_2 \in P_2\} \).

(b) Suppose that you are given the values \( \delta_L \) and \( \delta_R \) and each of the sets \( P_L \) and \( P_R \) sorted by \( y \)-coordinate. Show how to calculate \( \delta(P) = \min(\delta_L, \delta_R, d(S_L, S_R)) \) in \( O(n) \) time.

Hint. In \( O(n) \) time first find \( S_L \) and \( P_L \), each sorted by \( y \) coordinate. Then show how, in \( O(|S_L| + |S_R|) \) time, you can find \( d(S_L, S_R) \) by using the ideas from the gridding lemma.

(c) Now construct a divide and conquer algorithm for finding \( \delta(P) \) that works by

(i) Finding the median by \( x \)-coordinate of \( P \). Set this \( x \) coordinate to be \( X \).

(ii) Split \( P \) on \( x \) into \( P_L \) and \( P_R \).

(iii) Recusively find \( \delta(P_L) \) and \( \delta(P_R) \)

(iv) Use the ideas above to find \( \delta(P) \) using \( O(n \log n) \) extra time

Note that the recursion will terminate when \( P = \{p\} \) or \( P = \{p, p'\} \). In those cases \( \delta(P) = \infty \) or \( \delta(P) = d(p, p') \) can be found in \( O(1) \) time.

The correctness of the algorithm follows from (a) and (b).

Show how to implement the algorithm in \( O(n \log^2 n) \) time.

(d) Can you improve this to \( O(n \log n) \) time?