Instructions.

1. You could submit a pdf of the homework, either printed or hand-written and scanned, as long as it is easily readable.

2. If your solution is not clearly written, it might not be graded.

3. Prove the correctness of your answer. A correct answer without a proof might not be accepted.

4. If you have discussed the solution with other students, mention their names clearly on the homework. These discussions are not forbidden and are actually encouraged. However, you must write your whole solution yourself.

5. All questions have same weight.

6. You could refer to a data structure studied in class, and just mention briefly their guaranties. For example “It is known that a Red-Black tree could support the insert/delete/find operations on a set of n elements in time $O(\log n)$.”

7. If your answer uses one of the data structures or algorithms that were studied in class, you could refer to it without having to repeat details studied in class. If your answer requires only modifications of one of the algorithm, it is enough to mention the required modifications, and what’s the effect (if any) on the running time and on other operations that the algorithm performs.

8. In general, a complete answer should contain the following parts:
   (a) High level description of the data structures (if needed). E.g. We use a binary balanced search tree. Each node contains, a key and pointers to its children. We augment the tree so each node also contains a field...
   (b) High level description of the algorithms
   (c) Proof of correctness (why your algorithm provides what is required).
   (d) A claim about the running time, and a proof showing this claim.

9. When discussing trees, do not confuse the children of a node with the descendants of the node.
In all questions where the term projection on a line is mentioned, refer to the orthogonal projection as discussed in class.

1. Consider the process of inserting $m$ keys into a hash table $T[0..m - 1]$, where $m$ is a prime, and we use double addressing. The hash function we use is

$$h(k, i) = (k + i) \mod m$$

Give an example of $m$ keys $k_1, k_2 \ldots k_m$, such that the sequence of operations

- insert($k_1$)
- insert($k_2$)
  
  :  
- insert($k_m$)

Takes $\Omega(m^2)$ time.

2. Let $S = \{p_1 \ldots p_n\}$ be a set of $n$ points in the plane. Each point $p_i$ is given by its $x$ and $y$ coordinates $(x_i, y_i)$. Suggest a data structure that supports the following operations:

- Insert($x_i, y_i$). Insert a new point whose coordinates are $(x_i, y_i)$ into $S$.
- Sum($q$) returns the sum of the $y$-coordinates of all the points of $S$ whose $x$ coordinates are strictly smaller than $q$. That is, the sum of $y$-coordinates of the points of $S$ which are to the left of the vertical line $x = q$. If there is no such point, return zero.

The time for each operation should be $O(\log n)$.

**Example:** For the set of points in the figure,

- Sum(2) returns 3
- Sum(4) returns $3 + 1 = 4$. The line $x = 4$ is highlighted, and the points to its left are $p_1$ and $p_2$.
- Sum(6) returns $3 + 1 + 2 = 6$,
- Sum(8) returns $3 + 1 + 2 + 4 = 10$
3. (a) What is the projection (in 2D) of the point \((a, 2a)\) on the line passing through the points \((0, 0)\) and \((0, 1)\)? Your answer should depend on the unspecified value \(a\).

(b) What is the projection (in 2D) of the point \((a, 2a)\) on the line passing through the points \((0, 0)\) and \((2, 3)\)? Your answer should depend on the unspecified value \(a\).

(c) What is the projection point of the point \((a, 2a, 3a)\) on the line passing through the points \((0, 0, 0)\) and \((1, 1, 1)\)? Your answer should depend on the unspecified value \(a\). Note that a unit vector from \((0, 0, 0)\) in the direction of \((1, 1, 1)\) is

\[
\left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right)
\]

4. Consider the points \(p = (0, 0)\) and \(q = (2, 0)\). Let \(\ell\) be a line, passing through \((0, 0)\) and creating an angle \(\beta\) with the positive \(x\)-axis. Let \(\pi(p)\) and \(\pi(q)\) be the orthogonal projections of \(p\) and \(q\) on \(\ell\). Let \(d(\beta)\) be the distance from \(\pi(p)\) to \(\pi(q)\) (note that this is a function of \(\beta\)).

![Figure 1: Note that \(\pi(p)\) is the same point as \(p\).](image)

(a) for which values of \(\beta\) does \(d(\beta) = 2\)

(b) for which values of \(\beta\) does \(d(\beta) = 0\)

(c) Show what are the values of \(\beta\) for which \(d(\beta) < 1\) ?

(d) Conclude for the last item that if \(\beta\) is picked randomly, uniformly between 0° and 180°, then the probability that \(d(\beta) \geq 1\) is larger than the probability that \(d(\beta) \leq 1\).

5. The question refers to the algorithm for finding the closest pair, among a set of points in 2D. Let \(n\) be an arbitrary input. Suggest an example of a set \(S = \{p_1 \ldots p_n\}\) of \(n\) points, and the order in which they are processed by the algorithm, such that the running time of the algorithm is \(\Omega(n^2)\). For convenience, assume that all the points of \(S\) are on the \(x\)-axis.

6. Under the same assumptions as in the previous question, now assume that you pick a random permutation of \(S\). What is the probability that

\[
d(S_2) > d(S_3) > d(S_4) > \ldots > d(S_n).
\]

Assume that no two pairs of points from \(S\) define the same distance between them. That is

\[
\forall a, b, c, d \in S, \quad \text{dist}(a, b) \neq \text{dist}(c, d).
\]
7. Let $P$ be a set of points in the plane. Suggest an algorithm that finds in expected time $O(n)$ the triangle $\triangle p_1 p_2 p_3$ with smallest perimeter, among all triangles whose vertices are points of $P$. The perimeter is the sum of lengths of edges of the triangles. Your algorithm might run in $\Theta(n^2)$ in the worst case.

You could assume that no two triangles have the same perimeter. The algorithm is actually faster if this is not the case, but you are not required to prove it.