Please type up short answers to the following four questions. Formal and concise is best, but concise is more important than formal.

Email your answers to me (retep.dixon@utoronto.ca) by Saturday Nov 29 at noon.

1. The main TA roles for this course are:
   - Grader (grade assignments/exams, provide feedback)
   - Lecture TA (answer questions during in-class worksheets)
   - Tutorial TA (run tutorial sessions)

   Are there any roles you are not willing to do? Grading is the only one that can be done remotely. (There will be assorted hours for other tasks like invigilation or piazza.)

2. Imagine a student asks: "Since big O is worst case, is big Ω best case?"

   Answer their question in at most two paragraphs.

3. Explain the error in this proof:

   Definitions: Let $G = \langle V, E, w\rangle$ be an undirected, connected, weighted graph.

   A **path** in G is a sequence of vertices $v_1, \dots v_k$ where $\langle v_i, v_{i+1}\rangle \in E$ for all $1 \leq i < k$. We say the path is a **path from** s **to** t if $v_1 = s, v_k = t$. The **length of the path** is $\sum_{i=1}^{k-1} w(\langle v_i, v_{i+1}\rangle)$

   A **spanning tree** T of G is a connected, acyclic subgraph of G. The **cost** of T is $\sum_{e\in T} w(e)$. A **minimum spanning tree** M of G is a spanning tree satisfying: for any spanning tree T, cost$(M) \leq$ cost(T).

   Claim: A minimum spanning tree of a graph always contains the shortest path from s to t.

   Proof: Let M be a MST of G. Let p be a shortest path from s to t in G, and let q be the path from s to t in M. Suppose length(p) $\leq$length(q).

   Then $p \neq q$, so there is an edge in q that is not in p. Remove that edge. M is now disconnected.

   Because p is a path from s to t, there must be at least one edge in p that will connect M, so add that edge to M to create a new minimum spanning tree M'.

   Because length(p) $\leq$length(q), cost(M') $\leq$cost(M). This is a contradiction with the fact that M is a minimum spanning tree of G.

4. You have a data structure D that supports three operations: create, insert and merge.

   Create() returns an empty D and takes $O(1)$ time.

   Insert(D, x) runs in $O(\log n)$ time, where n is the number of items in D, and it increases the size of D by 1.

Merge(D_1, D_2) runs in $O(\min(n, m))$ time, where n is the number of items in D_1 and m is the number of items in D_2. Merge(D_1, D_2) deletes D_1 and D_2 and creates a new D_3 with size $n + m$.

Give a O-bound on the amortized runtime of n of these operations. A formal proof is not necessary, just describe the worst sequence, explain why it's the worst, and analyze the runtime.