Comp3121 assignment 2 Question 3
Chen En (z5335039)

Question 3

[20 marks] Ryno needs your help! He has two directed acyclic graphs X and Y. Each of these graphs have n vertices, labelled $1, \ldots, n$. He wants to know whether there is an ordering of $[1, \ldots, n]$ which is a topological ordering of ***both*** graphs, and if so, what that ordering is.

A trivial example where this ordering does *not* exist is when $n = 2$, and X has an edge between $(1, 2)$ and Y has an edge between $(2, 1)$.

3.1 **[5 marks]** Provide another *non-trivial* example where this ordering does *not* exist. Non-trivial in this case means that an edge cannot appear in one direction in X, and in the opposite direction in Y, as per the example earlier.

3.2 **[15 marks]** Design an $O(n \log n)$ algorithm which solves Ryno's problem.

3.1
Assume graph X and Y contain three vertices, we can take the graph X as 3->1->2 and graph Y as 1->2->3.

3.2

We can adapt the topological sorting algorithm mentioned in the lecture slide (greedy p24) on both graphs X and Y to solve this problem because X and Y are both directed acyclic graphs.

We use an empty list L to record the ordered elements and put all the vertices of X with no incoming edge into a set S. Then we check if there is an element in S also a vertex of Y with no incoming edge. If there is then we add this vertex u (i.e. a number between 1 and n) into L and we remove all the edges satisfying the condition: {e = (u, v) | v has no incoming edge on both X and Y}. Nevertheless, if we cannot find such a vertex u, that means we cannot find a topological sorting on both graphs. We continue the step above until S is empty. If either graph has edges left, that also means we cannot find a topological sorting on both graphs. Otherwise, the list L (a tropological sorted order) is the answer.

Time complexity

The worst case happens when we have to check every element in S and the maximum size of S is n. Also, for each iteration the size of L increases by 1. That is, the worst case time complexity is O (n * n) which is O(n^2) .