Problem 1

In this problem we consider a directed graph $G = (V, A)$ with n vertices and m arcs.

(a) A toplogical sort of the vertices is an ordering of the vertices such that there is no edge from a vertex u to v if u is placed after v in the ordering.

Formulate an $O(m + n)$ algorithm that finds a topological sort of the vertices or decides that there is a directed circuit in G.

(b) Assume for this part that G is acyclic, *i.e.*, there exists no directed cycle in G. Show how the above ordering can be used to compute single source shortest paths in a single pass using the Bellman-Ford algorithm.

(c) We will now see how the previous observation can be used to reduce the total number of iterations of the main loop in Bellman-Ford. The idea is to take an arbitrary vertex ordering $(v_1, \ldots, v_n)$ and split the edge set E into two sets $E_1 = \{v_iv_j \mid i < j\}$ and $E_2 = \{v_iv_j \mid i > j\}$. Both, $G_1 := (V, E_1)$ and $G_2 := (V, E_2)$, are directed acyclic graphs. How can you make use of this observation to reduce the number of iterations of the main loop in Bellman-Ford?

Problem 2

Given n numbers $a_1, \ldots, a_n$ find indices i and j, $1 \le i \le j \le n$, such that $\sum_{k=i}^{j} a_k$ is minimized. We will develop two algorithms for this problem that run in linear time, *i.e.*, the number of operations is linear in n.

(a) Solve the problem using Bellman-Ford as a subroutine. In particular, construct a graph such that a shortest path in this graph yields the optimal solution to the above problem. Show that the graph can be generated in linear time and that Bellman-Ford can be implemented to run in linear time on this graph.

(b) Define $d(j) = \min_{1 \le i \le j} \sum_{k=i}^{j} a_k$. Conclude that the above problem is equivalent to computing $\min_{1 \le j \le n} d(j)$. Show how this can be done in linear time.