Problem 1
Given \( n \) numbers \( a_1, \ldots, a_n \) find indices \( i \) and \( j \), \( 1 \leq i \leq j \leq n \), 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 (arithmetic) 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 \( d(j) = \min_{1 \leq i \leq j} \sum_{k=i}^{j} a_k \). Conclude that the above problem is equivalent to computing \( \min_{1 \leq j \leq n} d(j) \). Show how this can be done in linear time.

Problem 2
Due to the decentralized nature of the global currency market, it might be the case that an individual or an organized group makes a large profit without risk. Arbitrage is a phenomenon that refers to cases when it is possible to convert one unit of a currency into more than one unit of the same currency by using discrepancy in exchange rates. For example, consider the case that 1 CHF buys 60 RUB, 1 RUB buys 0.019 USD and 1 USD buys 0.93 CHF. This means that a trader can transform 1 CHF into \( 60 \cdot 0.019 \cdot 0.93 = 1.0602 \) CHF gaining a profit of 6.02%.

Given a list of currencies \( r_1, \ldots, r_n \) and a matrix \( E \in \mathbb{R}^{n \times n} \) where \( E_{i,j} \) specifies the exchange rate between currencies \( r_i \) and \( r_j \), design a polynomial time algorithm to test if there is a possibility of arbitrage. While modelling the problem, bear in mind that testing if a weighted directed graph has a negative cycle can be done in polynomial time.

Problem 3
Let \( D = (V, A) \) be a directed graph, \( w : A \rightarrow \mathbb{R} \) be arc weights and \( s \in V \). Suppose that there exists a path from \( s \) to each other node of \( V \).

Consider the following linear program:

\[
\begin{align*}
\max \quad & \sum_{v \in V \setminus \{s\}} x_v \\
\text{s.t.} \quad & x_v - x_u \leq w(u, v), \quad \forall (u, v) \in A \\
& x_s \leq 0.
\end{align*}
\]

(1)

Show the following:

a) This LP is feasible if and only if \( D \) has no negative cycle;

b) If \( D \) has no negative cycle, then (1) has a unique optimal solution.
Problem 4
Design an algorithm that, a directed graph \( G = (V, A) \), finds the number of shortest paths from \( s \) to \( t \) in time \( O(|V| + |A|) \).

Problem 5
A 2-matching in a graph is a collection of disjoint cycles that covers all the vertices. Show that a 2-matching can be computed in polynomial time, if such one exists. Note that it is allowed to pick an edge twice in a 2-matching, i.e., one can have a 2-cycle.

Hint: One may reduce the problem to finding a perfect matching in a bipartite graph.