Question 2.

This question is related to Dijkstra's algorithm that we saw in the class and has two parts. Recall that in the class we showed that Dijkstra's algorithm computes single source shortest paths in directed graphs where all the edge weights are non-negative.

1. **(2 points)** Construct a graph on at most 4 vertices with at least one edge of negative weight, such that when invoked on this graph with some source vertex s, Dijkstra's algorithm correctly computes the shortest paths from s to each of the vertices in the graph.
2. **(2 points)** Construct a graph on at most 4 vertices, such that when invoked on this graph with some source vertex s, Dijkstra's algorithm makes an error while computing shortest paths from s.

In both the cases, give clear, complete and succinct description of your constructions.

Question 3.

Recall the definition of a tree: a tree is an undirected graph that is connected and acyclic. Prove the following properties of trees.

1. **(3 points)** Prove that a tree on n vertices has exactly $n-1$ edges.
2. **(3 points)** Prove that an undirected graph is a tree if and only if there is a unique path between every pair of vertices.

Recall that an undirected graph is said to be simple if there is at most one edge between any pair of vertices and there are no self loops. The degree sequence of a graph is the sequence of degrees of the vertices in the graph, with these numbers put in non-increasing order.

The goal of this question is to design an algorithm that takes as input a finite list of non-negative integers in non-increasing order and decides if this list is **graphic,** i.e. there is a simple graph such that its degree sequence is exactly this list.

1. **(2 points)** Is the sequence $(6, 5, 5, 4, 3, 2, 1)$ graphic (Yes/No) ? No explanation is necessary.
2. **(4 points)** As a first step towards such an algorithm, show that a finite list $A = (s, t_1, \ldots, t_s, d_1, d_2, \ldots, d_n)$ is graphic if and only if the list A' obtained by arranging the list $B = (t_1 - 1, t_2 - 1, \ldots, t_s - 1, d_1, \ldots, d_n)$ in non-increasing order is graphic. Note that the length of A' is strictly smaller than the length of A.
3. **(4 points)** Using the second part of this question, design an efficient algorithm to decide if a given input list is graphic. Even if you haven't solved the second part of this question, you can still use the statement for answering this part.
4. **(2 point)** Prove the correctness of your algorithm in part 3.

Question 5.

Let G be a flow network defined as follows.

1. The vertices are s, t, u, v, w, x, where s and t are the source and the sink respectively.
2. The directed edges are (s, x), (s, v), (s, u), (x, w), (x, t), (v,u), (v,w), (w,t), (u,t).
3. The capacity of the edge (v,u) and (v,w) are 1 and that of the edge (x,w) is $\phi = \frac{\sqrt{5}-1}{2}$. Every other edge has capacity 10.

We now run the Ford-Fulkerson algorithm on this network G. The s-t path for each of the first few augmentations of the corresponding residual networks are picked as follows.

- Iteration 1: (s, v, w, t)
- Iteration 2: (s, x, w, v, u, t)
- Iteration 3: (s, v, w, x, t)
- Iteration 4: (s, x, w, v, u, t)
- Iteration 5: (s, u, v, w, t)

1. **(2 points)** What is the maximum value of a feasible s-t flow in G ? Clearly state the value of the flow function on each edge for this max flow.
2. **(8 points)** At the end of each of the five iterations specified above, state the value of the flow in G. To get clean expressions, you might have to use that the real number ϕ satisfies $\phi^2 + \phi - 1 = 0$ and is strictly less than 1.
3. **(2 points)** At the end of the fifth iteration, we again repeat the paths used above in this order and keep repeating paths in this order till the algorithm terminates. How many iterations would the Ford-Fulkerson algorithm take to terminate ? Give a short justification of your answer.