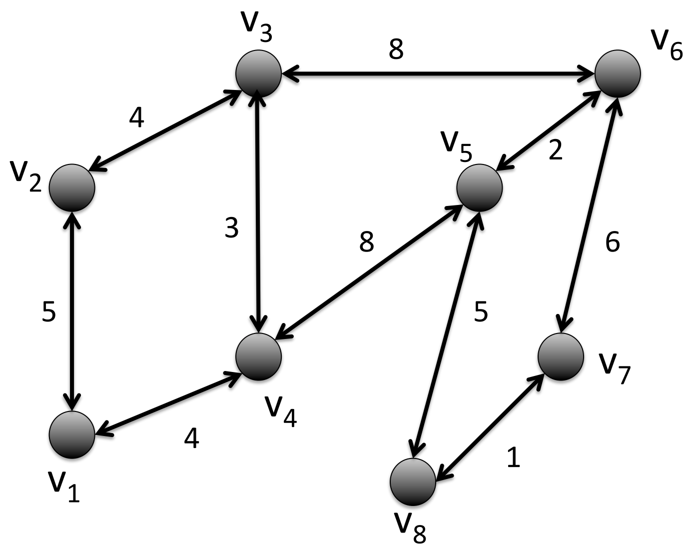
Shortest Paths and Minimum Spanning Trees

Exercise 1 *Kruskal's Algorithm*

Compute a minimum spanning for the following graph using Kruskal's algorithm. Show the status of your partial minimum spanning tree after each edge insertion and indicate for each edge whether it is included in the minimum spanning tree.



Exercise 2 *Single-source-shortest path problem in acyclic graphs*

We consider the single-source-shortest path problem of a given directed graph $G = (V, E)$ with non-negative edge weights and a source node s. Furthermore, we assume that the given graph G is acyclic, i. e. it does not contain a cycle.

- Give an algorithm (in pseudo-code) that solves the single-source-shortest path problem for a given acyclic graph in time $O(m + n)$.
- Prove the correctness of your algorithm.
- Explain why your algorithm runs in time $O(m + n)$.

Exercise 3 *Euler tours of directed graphs*

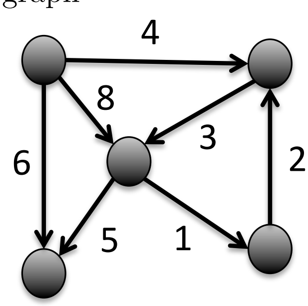
A Hamiltonian tour is a tour that visits every node of a strongly directed graph just once. Determining if a Hamiltonian tour exists is an NP-Complete problem.

In contrast an *Euler tour* of a strongly connected directed graph $G = (V, E)$ is a *cycle* that traverses each edge of G exactly once, although it may visit a node more than once. Answer the following questions:

- Show that G has an Euler tour if and only if the in-degree of every node v equals the out degree of v.
- Describe an $O(m)$ (where m is the number of edges in G) algorithm to find an Euler tour of G if such a tour exists (*Hint:* Merge edge-disjoint cycles).

Exercise 4 *Floyd-Warshall Algorithm*

Use the Floyd-Warshall algorithm to compute the all-pairs-shortest paths for the following graph



End of Questions