Homework 2

Deadline: 14:20 on December 12, 2025

Exercise 1. (10 pts) Given an unweighted, connected, undirected graph $G = (V, E)$, *an Euler Tour* in G is a cycle in G that visits all edges in E but doesn't visit any edge twice. Please design an efficient algorithm for finding a Euler tour in G. You can assume that G contains a Euler Tour.

Hint: You can start a DFS from any vertex $v \in V$. When the DFS discovers v again, you find a cycle. If the cycle is not a Euler tour, you can remove the cycle, and after the removal of the cycle, there must be a vertex on the cycle that still has edges, from which you can start a DFS again.

Exercise 2. (10 pts) Consider an undirected, weighted, connected graph $G = (V, E, w)$. Assume that all edge weights are integers in the range from 1 to W, and the value of W is given. Modify Kruskal's algorithm to improve the $O(|E| \log |V|)$ running time, and analyze the running time of your modification. Your modification should be an efficient one, and how efficient your modification should be is a part of the task.

Exercise 3. (15 pts) Consider an undirected, weighted, connected graph $G = (V, E, w)$ and three vertices $v_1, v_2, v_3 \in V$. Note that v_1, v_2 and v_3 are given, so you cannot arbitrarily choose three vertices. Assume that all edge weights are nonnegative. Design an efficient algorithm to find a subgraph $T = (V_T, E_T)$ of G with $V_T \subseteq V$ and $E_T \subseteq E$ such that (1) T is connected, (2) v_1, v_2, $v_3 \in V_T$, and (3) the total weight of E_T is minimized.

Hint: T consists of three paths or two paths. That is, T consists of three paths, P_1, P_2 and P_3 where $v_1 \overset{P_1}{\leftrightsquigarrow} s$, $v_2 \overset{P_2}{\leftrightsquigarrow} s$ and $v_3 \overset{P_3}{\leftrightsquigarrow} s$ for some vertex $s \in V$. If $s = v_i$, P_i is omitted, and thus T consists of two paths.

Exercise 4. (15 pts) Consider a directed, weighted graph $G = (V, E, w)$ without any negative cycle. As a convention for the all-pairs shortest paths problem, let $V = \{1, 2, \ldots, n\}$, let $D = \{d_{ij}\}$ be the distance matrix and let $\Pi = \{\pi_{ij}\}$ be the predecessor matrix. More precisely, d_{ij} is the length of a shortest path from i to j, and π_{ij} is the predecessor of j along a shortest path from i to j. Please explain how to compute D from Π and how to compute Π from D.
You have to analyze the two time complexities.

Exercise 5. (10 pts) Consider a directed, weighted graph $G = (V, E, w)$ with no negative cycle. Assume that V has a subset T such that for every ordered pair of vertices (u, v) of V, there is a shortest path from u to v whose intermediate vertices belong to T. Furthermore, T is given, $|T| = o(|V|)$, and $|E| = \Omega(|V^2|)$. Develop an efficient algorithm for computing all-pair shortest paths in G.

Exercise 6. (15 pts) You are given a directed graph $G = (V,\ E,\ w)$ with $w : E \mapsto \mathbb{R}_+$, a source $s \in V$ and a target $t \in V$. Two paths from s to t are called *vertex-disjoint* if they don't share any intermediate vertex, i.e., excluding the two endpoints s and t. Please design an efficient algorithm for finding a maximum set of vertex disjoint paths from s to t.
Hint: You can design a flow network by modifying G, e.g., modify all the vertices in $V \setminus \{s, t\}$.

Exercise 7. (15 pts) Consider a flow network $G = (V, E, c, s, t)$ and a positive integer k Assume that every edge in E has capacity 1 (i.e., $\forall e \in E$, $c(e) = 1$), and the distance from s to t in G is at least k. Prove that the minimum cut of G is at most $|E|/k$.
Hint: 1) Define $S_i = \{v \in V; d(s, v) \leq i\}$. 2) Prove that each edge $e \in E$ belongs to at most one of the sets $E\{S_i, V \setminus S_i\}$.

Exercise 8. (10 pts) As you have learned in the lecture, Ford–Fulkerson algorithm takes $O(EU)$ time for the maximum flow problem where E is the number of edges and U is the maximum capacity of an edge. The best paper award of FOCS 2022 is given to an $O(E^{1+o(1)} \log U)$-time algorithm by Chen, Kyng, Liu, Peng, Probst Gutenberg and Sachdeva. Both time complexities involve U. Please explain why the former one is not polynomial-time, but the latter one is polynomial-time.

Exercise 9. (Not graded) Given an undirected graph $G = (V, E)$, the *longest simple cycle* problem is to find a simple cycle in G with the maximum number of edges.

a Define a decision problem for the longest simple cycle problem.

b Prove that the decision problem belongs to NP.

c Prove that the decision problem is NP-Hard.

Hint: You can assume that the *Hamiltonian cycle* problem is NP-Complete. Recall that for an undirected graph $G = (V, E)$, a Hamiltonian cycle is a simple cycle that visits all vertices in V.

Exercise 10. (Not graded) Prove that the independent set problem can be reduced from the 3-CNF-SAT problem in polynomial time. You are not allowed to use the transitivity of the reduction.