Practice Problem Set: Dynamic Programming

Problem 1. Show that if any number of incoming arcs, with any capacities, are added to the source node, the maximum flow value remains unchanged. Similarly, show that if any number of outgoing arcs, with any capacities, are added at the sink node, the maximum flow value remains unchanged.

Problem 2. Several families go out to dinner together. To increase their social interaction, they would like to sit at tables in a way such that no two members of the same family are at the same table. Show how to formulate finding a seating arrangement that meets this objective as a maximum flow problem. Assume that there are p families and q tables where the i^{th} family has $c(i)$ members and the j^{th} table has a seating capacity of $c(j)$.

Problem 3. The following statements may or may not be correct. For each statement, provide either a short proof of correctness or a counter-example to show that it is incorrect.

1. The max-flow problem with node capacities is defined as follows. Given a directed graph $G = (V, E)$ with source s, sink t and capacities $c_v \geq 0$ for all nodes $v \in V$ (but no edge capacities c_e), a flow f is feasible in G if for all v other than s and t, the total flow into node v is at most c_v, and the size of a feasible flow f is the total flow out of s. Given this input, the max-flow problem with node capacities is to compute a feasible flow of maximum size.

2. Suppose a maximum (s,t)-flow f in a flow network G with integer capacities has already been computed. Let k be an arbitrary positive integer, and let e be an arbitrary edge in G whose capacity is at least k. Next, suppose the capacity of e is increased by k units. Then, the maximum flow in the updated graph can be computed in $O(|E|k)$ time.

3. If all edges in a graph have distinct capacities, there is a unique maximum flow.

4. If we add a positive number λ to all edge capacities, the minimum cut remains unchanged.

5. If we add a positive number λ to all edge capacities, the maximum flow increases by a multiple of λ.

Problem 4. Let $G = (V, E)$ be a directed graph, with source $s \in V$, sink $t \in V$, and non-negative edge capacities c_e. Give a polynomial-time algorithm to decide whether G has a unique minimum $s - t$ cut (i.e.,an $s - t$ of capacity strictly less than that of all other $s - t$ cuts).

Problem 5. Given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a designated source $s \in V$, and a designated sink $t \in V$, an integer maximum s-t flow in G is known and defined by a flow value f_e on each edge e. Then, suppose the capacity of a specific edge $e \in E$ is increased by one unit. Show how to find a maximum flow in the resulting capacitated graph in time $O(m + n)$, where $|V| = n$ and $|E| = m$.

Problem 6. Consider the following problem. You are given a flow network with unit-capacity edges: it consists of a directed graph $G = (V, E)$, a source s and a sink t, and capacity $c_e = 1$ for every edge e. You are also given a parameter k. The goal is delete k edges so as to reduce the maximum $s - t$ flow in G by as much as possible. In other words, you should find a set of edges $F \subset E$ so that $|F| = k$ and the maximum $s - t$ flow in the graph $G_0 = (V, E\ F)$ is as small as possible. Give a polynomial-time algorithm to solve this problem.