Design and Analysis of Algorithms

Practice-sheet: **Maximum Flow**

1. (**Flow fundamental**) Suppose you are given a directed graph $G = (V, E)$ with a positive integer capacity c_e on each edge, a designated source $s \in V$, and a designated sink $t \in V$. You are also given an integer maximum $s - t$ flow in G, defined by a flow value f_e on each edge e.

   Now suppose we pick a specific edge $e \in E$ and increase its capacity by one unit. Show how to find a maximum flow in the resulting capacitated graph in time $O(m + n)$, where m is the number of edges in G and n is the number of vertices in G.

2. **Blood bank problem**

   We all know the basic rule for blood donation: A patient of blood group A can receive only blood of group A or O. A patient of blood group B can receive only blood of group A or O. A patient of blood group O can receive only blood of group O. A patient of blood group AB can receive blood of any group.

   Let s_O, s_A,s_B, s_{AB} denote the supply in whole units of the different blood types on hand. Assume that the hospital knows the projected demand for each blood type d_O,d_A,d_B, and d_{AB} for the coming week. Give a polynomial time algorithm to evaluate if the blood on hand would suffice for the projected need. You should formulate this problem as a max-flow problem, establish a relation between the two problems by stating a theorem, and then you should prove the theorem.

3. (**Mobile phone and base stations**)

   Consider a set of mobile computing clients in a certain town who each need to be connected to one of several possible base stations. There are n clients with the position of each client specified by its (x, y) coordinates in the plane. There are also k base stations; the position of each of these is specified by (x, y) coordinates as well.

   For each client, we wish to connect it to exactly one of the base stations. Our choice of connections is constrained in the following ways. There is a range-parameter r - a client can only be connected to a base station that is within distance r. There is also a load parameter L - no more than L clients can be connected to any single base station.

   Your goal is to design a polynomial time algorithm for the following problem. Given the position of a set of clients and a set of base stations, as well as the range and load parameters, decide whether every client can be connected simultaneously to a base station, subject to the range and load conditions in the previous paragraph.

4. (**Max-damage to network**)

   You are given a flow network with unit capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter k.

The goal is to delete k edges so as to reduce the maximum-flow on G by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum $s - t$ flow in $G' = (V, E - F)$ is as small as possible subject to this. Give a polynomial time algorithm to solve this problem.

5. (**unique min-cut**)

   Let $G = (V, E)$ be a directed graph with source $s \in V$, sink $t \in V$ and nonnegative 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 cut of capacity strictly less than that of all other s-t cuts.)

6. (**Vertex disjoint paths**)

   There is a directed graph $G = (V, E)$ on n vertices and m edges. There are two vertices $s, t \in V$. Two paths from s and t are said to be vertex disjoint if they do not share any vertex except s and t Design a polynomial time algorithm to compute the maximum number of vertex disjoint paths from s to t.

7. (**Application with lower bound on flow**)

   There are n boxes: $B_1, \ldots, B_n$. Box B_i has three dimensions (ℓ_i, b_i, d_i) and each of them is in the range from 2 feet to 3 feet. We say that box B_j can be *placed inside* box B_i if each of the dimensions of B_j (possibly after appropriate rotation if needed) is less than the corresponding dimension of B_i. For example, box with dimensions (5,3,4) can be placed inside another box with dimensions (4,6,5). A box is no more visible once it gets placed inside another box. Design a polynomial time algorithm to place boxes into each others suitably so that the number of visible boxes is the lest possible.

8. (**Circulation with lower bound on flow**)

   Recall the circulation problem which we solved by reducing to max-flow problem. We shall now extend this problem further.

   There is a flow network $G = (V, E)$ with source, sink $t \in V$ and nonnegative edge capacities $\{c_e\}$. Each vertex v has a demand d_v which is a real number. In addition each edge has a nonnegative number ℓ_e. Design a polynomial time algorithm to determine if there exists a circulation $f : E \to R$ such that

   (a) For each vertex v, $f_{in}(v) - f_{out}(v) = d_v$.

   (b) For each edge e, $\ell_e \leq f(e) \leq c_e$.

   Hint: Reduce this problem to an instance of circulation problem without any lower bound on edges.

9. **Farthest Min-cut**

   There may be many min-cuts in a flow network from s to t. Min-cut C_0 defined by a pair (A_0, B_0) of vertex sets is said to be the farthest min-cut if for every other min-cut C', say defined by (A', B'), the subset B_0 must be contained in B'. Design an efficient algorithm to compute the farthest min-cut.

10. **Rounding**

    Let M a $n \times n$ matrix storing positive real numbers. However, sum of elements of each column is an integer and sum of elements of each row is also an integer. Let e be any real number. Rounding of e means replacing e by $\lfloor e \rfloor$ or $\lceil e \rceil$. Prove that elements of M can be rounded without changing any column sum or row sum.
    Note: We discussed this problem and its solution in the class. The arguments used were quite subtle. You are strongly recommended to reconstruct the solution on your own.

11. **Disaster management**

    Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.

    Consider the following scenario. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospital. There are k hospital in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals depending on where they are right now).

    At the same time, one does not want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is balanced: each hospital receives at most $\lceil n/k \rceil$ people.

    Give a polynomial time algorithm that takes the given information about the people's locations and determines whether it is possible.

12. **An amazing application of Min-cut**

    Suppose we are given a directed network $G = (V, E)$ with a root node r and a set of *terminals* $T \subseteq V$. We would like to disconnect many terminals from r, while cutting relatively few edges.

    We make this trade-off precise as follows. For a set of edges $F \subseteq E$, let $q(F)$ denote the number of nodes $v \in T$ such that there is no $r-v$ path in the subgraph $(V, E-F)$. Give a polynomial time algorithm to find a set F of edges that maximizes the quantity $q(F) - |F|$. (Note that setting F equal to the empty set is an option.)