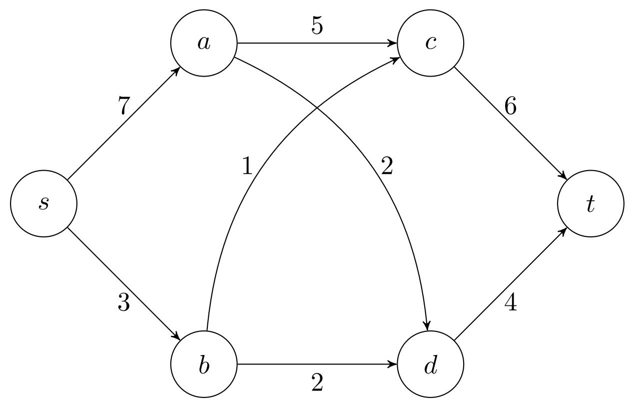
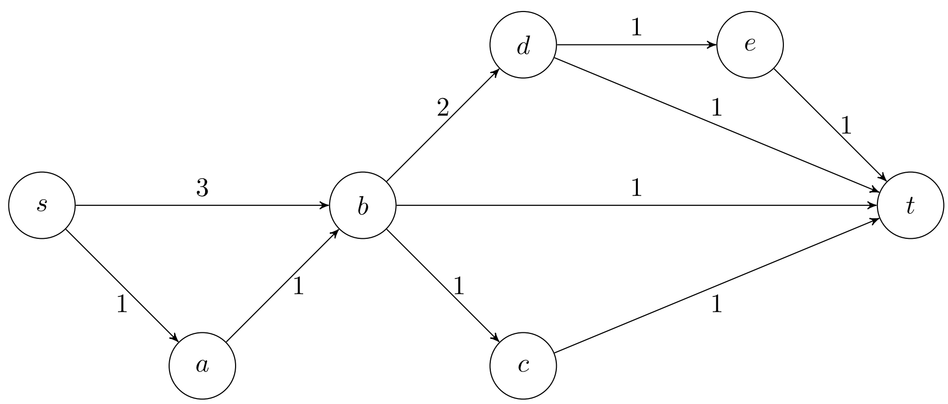
Weekly Assignment 6:
More Flow algorithms

1. Run the capacity scaling algorithm on the following flow network.



2. Run the Edmonds-Karp algorithm on the following flow network.



3. 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.

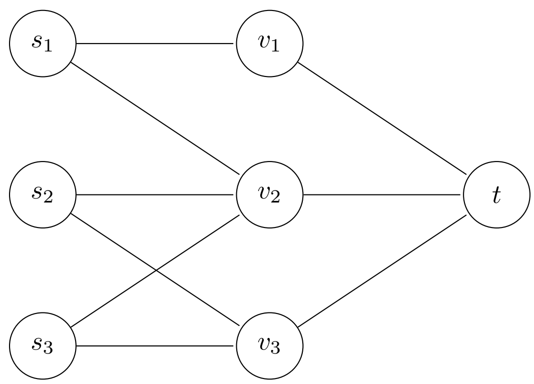
   We wish to connect each client to exactly one of the base stations. Our choice of connection is constrained in the following ways. There is a *range parameter* r — a client can only be connected to a base station that is within distance r. There is also a *load parameter* L — no more than L clients can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

Design a polynomial-time algorithm for the following problem. Given the positions 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 conditions from the previous paragraph.

4. Suppose, we have a graph $G = (V, E)$, a subset $S \subseteq V$ of source vertices, and a target vertex $t \in V$. A collection of **edge-disjoint paths** is a set of paths, such that:

   1. each of these paths must start in some $s \in S$ and must end in t; and
   2. none of these paths share an edge.

   Let us look at an example:



   Let $S = \{s_1, s_2, s_3\}$. Then the set consisting of the paths

$$s_1 \to v_1 \to t, \quad s_2 \to v_3 \to t, \quad s_3 \to v_2 \to t$$

   is edge-disjoint. However, if we add the path $s_1 \to v_2 \to t$, then it is not edge-disjoint anymore.

   In this exercise, we look the following problem: given G, S and t as above, and for each element $s \in S$ an natural number $k_s \geq 0$, is there a collection of edge disjoint paths such that for each $s \in S$, this collection contains at least k_s paths from s to t? Describe an algorithm that solves this problem and returns a boolean that indicates where such a collection exists. Provide an analysis of the complexity and correctness of your algorithm.

5. Consider the problem faced by a hospital that is trying to evaluate whether their blood supply for blood transfusions is sufficient.

   The basic rule for blood donation is the following. A person's own blood supply has certain *antigens* present (we can think of antigens as a kind of molecular signature); and a person cannot receive blood with a particular antigen if their own blood does not have this antigen present. Concretely, this principle underpins the division of blood into four *types*: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus, patients with type A can receive only blood types A or O in a transfusion, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.

   (a) Let s_O, s_A, s_B and 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.

   (b) Consider the following example. Over the next week, they expect to need at most 100 units of blood. The typical distribution of blood types in patients is 45% type O, 42% type A, 10% type

B, and 3% type AB. The hospital wants to know if the blood supply they have on hand would be enough if 100 patients arrive with the expected type distribution. There is a total of 105 units of blood on hand. The table below gives these demands, and the supply on hand.

blood type:	O	A	B	AB
supply:	50	36	11	8
demand:	45	42	10	3

Is the 105 units of blood on hand enough to satisfy the 100 units of demand? Find an allocation that satisfies the maximum possible number of patients. Use an argument based on a minimum capacity cut to show why not all patients can receive blood. Also, provide an explanation for this fact that would be understandable to the clinic administrators, who have not taken a course on Algorithms & Datastructures. (So, for example, this explanation should not involve the words "flow, " "cut, " or "graph" in the sense we use them.)