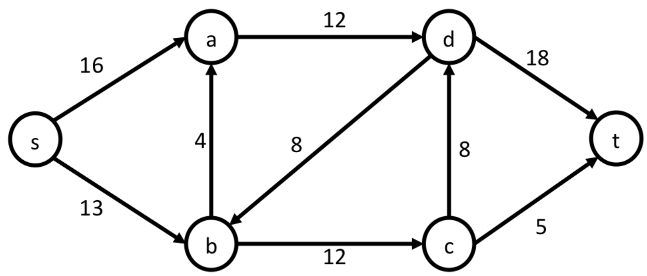
- The instructions are the same as in Homework-0, 1, 2.

There are 5 questions for a total of 100 points.

---

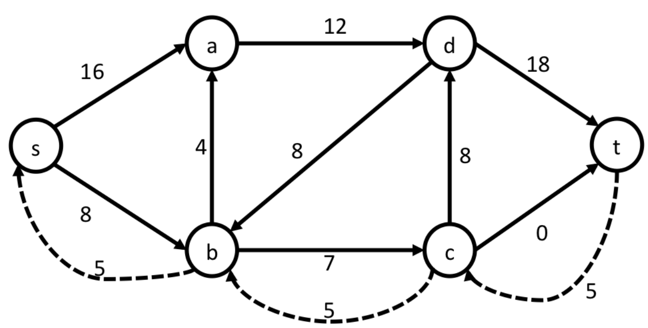
1. (20 points) Consider the network shown in the figure. Consider running the Ford-Fulkerson algorithm on this network.



(a) We start with a zero s-t flow f. The algorithm then finds an augmenting path in G_f. Suppose the augmenting path is $s \to b \to c \to t$. Give the flow f' after augmenting flow along this path.

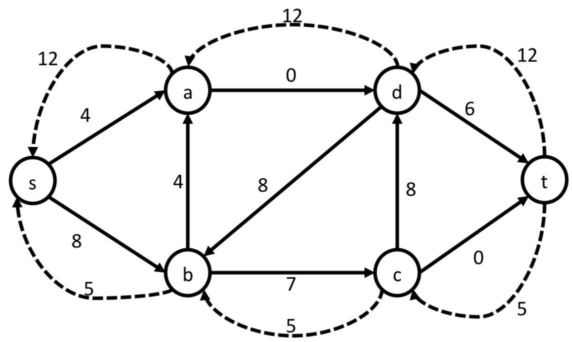
Solution: $f'(s,a) = 0, f'(s,b) = 5, f'(a,d) = 0, f'(b,a) = 0, f'(b,c) = 5, f'(c,d) = 0, f'(c,t) = 5, f'(d,b) = 0, f'(d,t) = 0$.

(b) Show the graph $G_{f'}$. That is, the residual graph with respect to s-t flow f'.



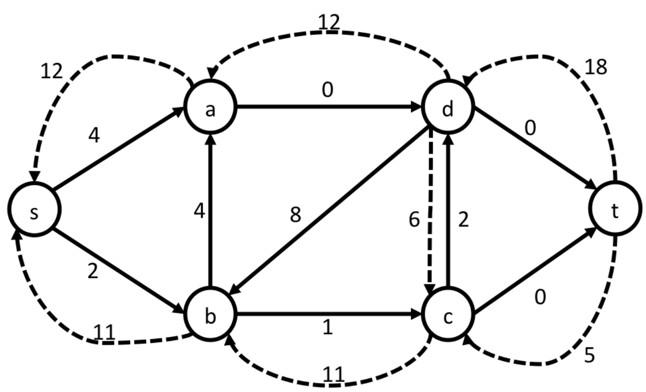
(c) The algorithm then sets f as f' and G_f as $G_{f'}$ and repeats. Suppose the augmenting path chosen in the next iteration of the while loop is $s \to a \to d \to t$. Give f' after augmenting flow along this path and show $G_{f'}$.

> **Solution:** $f'(s,a) = 12, f'(s,b) = 5, f'(a,d) = 12, f'(b,a) = 0, f'(b,c) = 5, f'(c,d) = 0, f'(c,t) = 5, f'(d,b) = 0, f'(d,t) = 12$. $G_{f'}$ is shown in the figure below.



(d) Let f be the flow when the algorithm terminates. Give the flow f and draw the residual graph G_f.

> **Solution:** $f(s,a) = 12, f(s,b) = 11, f(a,d) = 12, f(b,a) = 0, f(b,c) = 11, f(c,d) = 6, f(c,t) = 5, f(d,b) = 0, f(d,t) = 18$. G_f is shown in the figure below.



(e) Give the value of the flow f when the algorithm terminates. Let A^* be the vertices reachable (using edges of positive weight) from s in G_f and let B^* be the remaining vertices. Give A^* and B^*. Give the capacity of the cut (A^*, B^*).

> **Solution:** The value of the flow when the algorithm terminates is **23**. $A^* = \{s, a, b, c, d\}$, $B^* = \{t\}$ and the capacity of the cut (A^*, B^*) is **23**.

2. Answer the following:

(a) (1 point) State true or false: For every s-t network graph G, there is a unique s-t cut with minimum capacity.

(a) **False**

(b) (4 points) Give reason for your answer to part (a).

Solution: Consider a graph with 3 vertices s, a, t and edges $(s, a), (a, t)$ each of weight 1. $(\{s\}, \{a, t\})$ and $(\{s, a\}, \{t\})$ are both s-t cuts with minimum capacity.

(c) (1 point) State true or false: For every s-t network graph G and any edge e in the graph G, increasing the capacity of e increases the value of maximum flow.

(c) **False**

(d) (4 points) Give reason for your answer to part (c).

Solution: Consider the graph with 3 vertices s, a, t and edges (s, a) with weight 2 and (a, t) with weight 1. Increasing the capacity of the edge (s, a) in this graph does not increase the value of maximum flow.

(e) (1 point) State true or false: For every s-t network graph G for which an s-t flow with non-zero value exists, there exists an edge e in the graph such that decreasing the capacity of e decreases the value of maximum flow.

(e) **True**

(f) (4 points) Give reason for your answer to part (e).

Solution: For any s-t network graph G, consider an s-t cut (A, B) with minimum value in G. Consider any edge (u, v) such that $u \in A$ and $v \in B$. Decreasing the capacity of this edge decreases the capacity of the minimum cut in the graph which means that the value of maximum flow decreases.

3. Suppose you are given a bipartite graph (L, R, E), where L denotes the vertices on the left, R denotes the vertices on the right and E denote the set of edges. Furthermore it is given that degree of every vertex is exactly d (you may assume that $d > 0$). We will construct a flow network G using this bipartite graph in the following manner: G has $|L| + |R| + 2$ vertices. There is a vertex corresponding to every vertex in L and R. There is also a source vertex s and a sink vertex t. There are directed edges with weight 1 from s to all vertices in L and directed edges of weight 1 from all vertices in R to t. For each edge $(u, v) \in E$, there is a directed edge from u to v with weight 1 in G.

(*The figure below shows an example of a bipartite graph and the construction of the network.*)

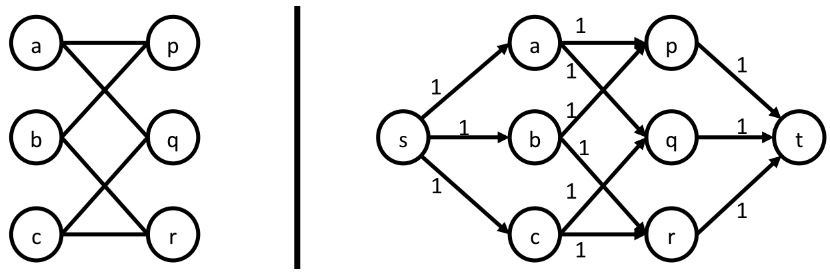


Figure 1: An example bipartite graph (with $d = 2$) and network construction.

(a) (5 points) Argue that for any such bipartite graph where the degree of every vertex is equal to d, $|L|$ is equal to $|R|$.

Solution: Since every vertex in L has degree $d > 0$, we have $|E| = d \cdot |L|$. Similarly, we have $|E| = d \cdot |R|$. So, we get $d \cdot |L| = |E| = d \cdot |R| \Rightarrow |L| = |R|$.

(b) (10 points) Argue that for any given bipartite graph where the degree of every vertex is the same non-zero value d, there is an integer s-t flow (i,e., flow along any edge is an integer) in the corresponding network with value $|L|$.

Solution: Consider the following s-t flow f:

- For every $u \in L$, $f(s, u) = 1$
- For every $v \in R$, $f(v, t) = 1$
- For every u, v such that $u \in L$ and $v \in R$, $f(u, v) = 1/d$

Note that both capacity constraints and flow conservation requirements are satisfied by this flow f. The value of the flow is $|L|$. Note that there cannot be a flow with more value than $|L|$. So, f is an s-t flow with maximum value. Note that Ford-Fulkerson returns a flow with maximum value. Moreover, since the capacities are all integers, it returns an integer flow with maximum value. This means that the Ford-Fulkerson algorithm returns an integer flow with value $|L|$. This implies that there is an integer flow with value $|L|$.

(c) (5 points) A matching in a bipartite graph $G = (L, R, E)$ is a subset of edges $S \subseteq E$ such that for every vertex $v \in L \cup R$, v is present as an endpoint of at most one edge in S. A maximum matching is a matching of maximum cardinality. Show that the size of the maximum matching in any bipartite graph $G = (L, R, E)$ is the same as the maximum flow value in the corresponding network graph defined as above.

4. (20 points) There are n stationary mobile-phones $c_1, ..., c_n$ and n stationary mobile-phone towers $t_1, ..., t_n$. The distances between mobile-phones and towers are given to you in an $n \times n$ matrix d, where $d[i, j]$ denotes the distance between phone c_i and tower t_j. It is possible for a mobile-phone c_i to connect to a tower t_j if and only if the distance between c_i and t_j is at most D, where D is the connecting radius. Furthermore, at one time, a mobile-phone can connect to at most one tower and a tower can allow at most one connection. Your goal as a Communications Engineer is to figure out whether all mobile-phones are usable simultaneously. That is, whether it is possible for all mobile-phones to connect simultaneously to distinct towers. Answer the following questions.

(a) Consider a simple example with 5 mobile-phones and 5 towers. Let the connecting radius be $D = 2$ miles. The distance matrix for this example is given below.

d	1	2	3	4	5
1	1	2	3	4	7
2	4	1	1	5	12
3	5	7	2	1	11
4	4	3	6	1	1
5	1	21	8	9	13

Figure 2: Distance matrix d for part (a) of question 5

Prove or disprove: It is possible for all 5 mobile-phones to simultaneously connect to distinct towers for this example.

Solution: The following simultaneous connections are possible:

- c_1 connects to t_2
- c_2 connects to t_3
- c_3 connects to t_4
- c_4 connects to t_5
- c_5 connects to t_1

(b) Design an algorithm that takes inputs n, D, and the distance matrix d, and outputs "yes" if it is possible for all mobile-phones to simultaneously connect to distinct towers (within the connecting radius D) and "no" otherwise. Analyze the running time of the algorithm and give proof of correctness.

Solution: Here the pseudocode for the algorithm:

`SimultaneousConnection`(n, D, d)
- Construct a bipartite graph $G = (L, R, E)$ where $L = R = \{1, ..., n\}$ and $(i, j) \in E$ if and only if $d[c_i, t_j] \leq D$
- Use the algorithm discussed in class to find the maximum matching in G
- If the maximum matching in G is of size n, then return("yes") else return("no")

Running time: Creating the bipartite graph will take time $O(n^2)$ since we will have check the distance between c_i and t_j for all pairs (i, j). Finding the maximum matching in a bipartite graph is through the Ford-Fulkerson algorithm and we know that the running time of the algorithm is $O(m \cdot C)$ where m is the number of edges in the network graph on which Ford-Fulkerson is run and C is the sum of weight of edges from s. Note that in this case, $m = O(n^2)$ and $C = O(n)$. So, the running time of the Ford-Fulkerson algorithm is $O(n^3)$. So the total running time of the algorithm is $O(n^3)$.

5. (25 points) Town authorities of a certain town are planning for an impending virus outbreak. They want to plan for panic buying and taking cues from some other towns, they know that one of the items that the town may run out of is toilet paper. They have asked your help to figure out whether the toilet paper demand of all n residents can be met. They provide you with the following information:

   - There are n residents $r_1, ..., r_n$, m stores $s_1, ..., s_m$, and p toilet paper suppliers $x_1, ..., x_p$.
   - The demand of each of the residents in terms of the number of rolls required.
   - The list of stores that each of the residents can visit and purchase rolls from. A store cannot put any restriction on the number of rolls a customer can purchase given that those many rolls are available at the store.
   - The number of rolls that supplier x_j can supply to store s_i for all $i \in \{1, ..., m\}$ and $j \in \{1, ..., p\}$.

   The above information is provided in the following data structures:

   - A 1-dimensional integer array $D[1...n]$ of size n, where $D[i]$ is the demand of resident r_i.
   - A 2-dimensional 0/1 array $V[1...n, 1...m]$ of size $n \times m$, where $V[i, j] = 1$ if resident r_i can visit store s_j and $V[i, j] = 0$ otherwise.
   - A 2-dimensional integer array $W[1...m, 1...p]$ of size $m \times p$, where $W[i, j]$ is the number of rolls of toilet paper that the supplier x_j can supply to store s_i.

   Design an algorithm to determine if the demand of all residents can be met. That is, given (n, m, p, D, V, W) as input, your algorithm should output "yes" if it is possible for all residents to obtain the required number of rolls and "no" otherwise. Argue correctness and discuss running time.

   (*For example, consider that the town has two residents, one store and two suppliers. If* $D = [2, 2], V = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, *and* $W = [2, 2]$, *then the demand can be met. However, if* $D = [2, 2], V = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, *and* $W = [2, 1]$, *then the demand cannot be met.*)

Solution: We will design a network flow based algorithm for this problem. Given an instance (n, m, p, D, V, W) of this problem we will construct a network graph such that the toilet paper demand of all residents of the town can be met if and only if the maximum s-t flow in the network is $d = \sum_{i=1}^{n} D[i]$. The description of the algorithm is given in the pseudocode below:

`ToiletPaperDemand`(n, m, p, D, V, W)
- Construct a flow network $G = (R \cup S \cup X \cup \{s, t\}, E)$ in the following manner:
  - There is a node for every resident. So, $R = \{r_1, ..., r_n\}$.
  - There is a node for every store. So, $S = \{s_1, ..., s_m\}$.
  - There is a node for every supplier. So, $X = \{x_1, ..., x_p\}$.
  - There is a source node s and a sink node t.
  - For every $k \in \{1, ..., p\}$, $(s, x_k) \in E$. The capacity of this edge is ∞.
  - For every $k \in \{1, ..., p\}$ and $j \in \{1, ..., m\}$, $(x_k, s_j) \in E$. The capacity of edge is $W[j, k]$.
  - For every $j \in \{1, ..., m\}$ and $i \in \{1, ..., n\}$, $(s_j, r_i) \in E$.
    The capacity of this edge is ∞ if $V[i, j] = 1$ and 0 otherwise.
  - For every $i \in \{1, ..., n\}$, $(r_i, t) \in E$. The capacity of this edge is $D[i]$.
- Run `Ford-Fulkerson` algorithm to find a maximum s-t flow f in graph G.
- If $(v(f) == \sum_{i=1}^{n} D[i])$ then return("yes") else return("no").

Running time: Let $d = \sum_{i=1}^{n} D[i]$. Constructing the graph G takes $O(nm + mp)$ time since we need to go through the arrays V, W. Note that the number of edges in the graph G is $O(nm + mp)$.

The running time of `Ford-Fulkerson` will be $O(d \cdot (nm + mp))$. So the total running time is $O(d \cdot (nm + mp))$.

Correctness: Let G be the graph constructed in the first step of the above algorithm. The correctness follows from the two claims below:

Claim 1: If it is possible to meet all the demands, then there is a flow of value d in G.

Proof sketch. Since all the demands are met, there is a way in which each resident is able to acquire exactly $D[i]$ rolls of paper and each store sells off all its supplies (*any excess may be interpreted as returned to supplier which in turn may be interpreted as smaller supply from suppliers*). Let us consider this scenario and build a flow in the graph G. The flow along edges (r_i, t) is exactly $D[i]$ for every i. The number of rolls purchased by r_i at store s_j is the flow along the edge (s_j, r_i) and the number of rolls supplied by x_k to store s_j is the flow along edge (x_k, s_j). This is a valid s-t flow since capacity constraints and flow conservation is met. □

Claim 2: If there is a flow of value d in G, then it is possible for the toilet paper demand to be met.

Proof sketch. Given a flow f of value d, here is a way to meet all the demand: The flow along every edge (s_j, r_i) is the number of rolls that the resident r_i purchases at store s_j. The flow along every edge (x_k, s_j) is the number of rolls supplier x_k supplies to store s_j. Since the flow value is exactly equal to d, the number of rolls purchased by resident r_i as per the above scenario is exactly equal to $D[i]$. Moreover, due to flow conservation, a store cannot sell more than its supply and due to capacity constraint, a supplier cannot supply more than its capacity to any store. □