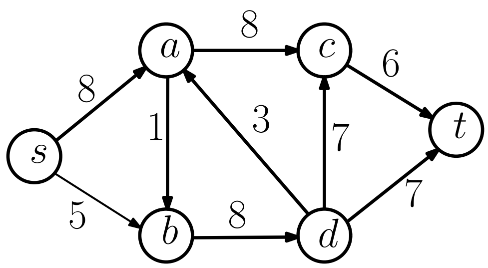
15-451/651 Algorithm Design & Analysis
Spring 2023, Recitation #6

Objectives

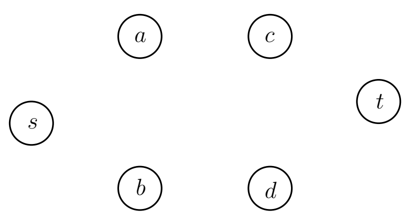
- Understand max s-t flows and min s-t cuts, and how to model problems with them

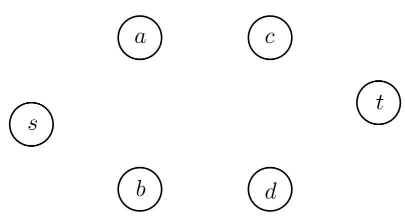
Recitation Problems

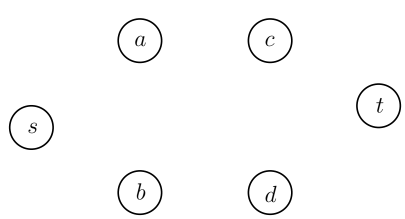
1. **(Ford-Fulkerson Practice)** Consider the graph below.



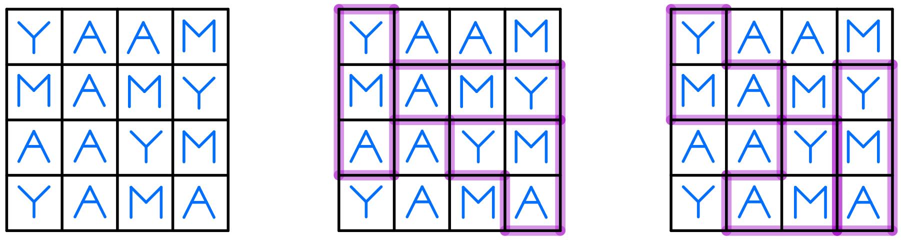
Run Ford-Fulkerson, drawing the **residual network** at every step, and show the final residual network. What is the maximum flow? What is the capacity of the minimum cut? Partition the vertices into two sets (S, T) such that $s \in S, t \in T$, and (S, T) is a minimum cut.







2. **(Amy Tiles)** There is an n by n grid of the letters A, M, and Y. The goal is to form as many disjoint copies of the word AMY as possible in the given grid. To form AMY you start at any A, move to a neighboring M, then move to a neighboring Y. (You can move north, east, south or west to get to a neighbor.) The following figure shows one such problem on the left, along with two possible solutions with three AMYs the right.



Give a polynomial-time algorithm to find the solution with the largest number of AMYs. Your algorithm should consist in reducing the problem to an instance of max flow.

3. **(Cut into Teams)** CMU intramural sports season is coming up, and Maximus Flow and Minnie Cut want to form teams to compete in the intramural football game. They have a group of n students that they want to split into two teams with Maximus captaining team A and Minnie captaining team B. Each student has some preference toward being on Maximus' or Minnie's team however. In particular, Max and Min need to pay \$$A_i$ for student i to join team A and \$$B_i$ for student i to join team B. To further complicate things, some pairs of students are friends. For each friend pair, they would prefer to be on the same team. For each friend pair (i, j) that ends up on different teams, Max and Min will have to pay an extra \$$K_{i,j}$ for making them unhappy. Being college students with no money and no free time, Maximus and Minnie want to form these teams with the lowest price as possible and in polynomial time. Note that the two teams do not have to be of the same size.

   (a) Reduce this to constructing a flow network where a given cut (partition of the vertices into two sets (A, B)) corresponds to assigning students to either team A or B.

   (b) Now suppose each friend pair (i, j) refused to be separated and will quit if they do. How can you adjust your flow network without deleting/merging any nodes to guarantee that no such friend pair will be put on different teams?

Further Review

1. **(Net flow)** Recall that the *net flow* across an s-t cut (S, T) is defined to be

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u),$$

   and that the *value* of a flow is defined to be

$$|f| = \sum_{u \in V} f(s,u) - \sum_{u \in V} f(u,s).$$

   Prove formally, using the definitions, that the net flow across *any s-t cut* is $|f|$.

2. **(Fractional max flow)** Give an example of a flow network and a maximum flow on that network such that the capacities are all integers, but the flow contains some non-integer values.

3. **(Multi-source multi-sink max flow)** Consider a variant of the maximum flow problem where we allow for *multiple source vertices* and *multiple sink vertices*. All of the sources and sinks are excluded from the flow conservation constraint. Describe a simple method for solving this problem by reducing it to an instance of the ordinary problem.

4. **(Vertex Capacities)** The usual formulation of the max flow problem has capacities on the *edges* of the network, but it is often very useful when modeling problems to be able to also have *vertex capacities*. That is, we want to limit the amount of flow that can pass through a particular vertex. Formally, if a vertex v has capacity $c(v)$, then we wish to enforce that the flow f satisfies

$$\sum_{u} f(u,v) \leq c(v).$$

   (Note that by flow conservation, this is equivalent to $\sum_u f(v,u) \leq c(v)$.) Devise a transformation that takes as input a max flow problem with vertex capacities and produces an instance of the ordinary max flow problem whose solution corresponds to a valid solution of the original problem.

5. **(Square sums)** Given a list of n integers $r_1, \ldots r_n$ and m integers $c_1, \ldots, c_m$, we want to construct an $n \times m$ matrix consisting of zeros and ones whose row sums are $r_1, \ldots r_n$ respectively and whose column sums are $c_1, \ldots, c_m$ respectively. Describe a solution to this problem by reducing it to an instance of maximum flow.

6. **(Office hours)** You are in charge of assigning TAs to 15-451's office hours. There are n TAs and m office hours slots on the calendar. The i^{th} TA has a list of which office hour slots they are available for, and also a limit c_i of how many slots they are willing to staff. Since different office hour slots have different numbers of students, each might require a different number of TAs. The j^{th} office hour slot requires d_j TAs to staff it.

   Give an algorithm for finding a viable schedule of which TAs staff which recitations, or determine that it is not possible. Do so by reducing the problem to max flow.

7. **(Edge-disjoint Paths)** We are given a directed graph G and two vertices u and v in G. We want to find the maximum possible number of edge-disjoint paths from u to v in G, meaning any edge in G can be used by at most one path.

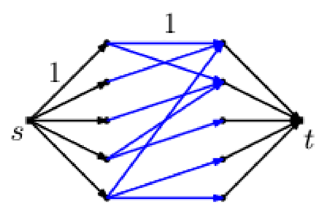
   (a) Solve this problem by constructing a flow network whose max flow is the number of edge disjoint paths.

   (b) Prove that the max flow of this network is equal to the maximum number of edge disjoint paths.

   (c) Given a max flow in the flow network you constructed, how would you construct a maximum set of edge-disjoint paths?

8. **(Hall's marriage theorem)** Given a bipartite graph $G = (L, R, E)$, we want to find the maximum matching in G. Recall from lectures that the following reduction allows us to solve this problem using maximum flow.

   - Add a new source vertex s and target vertex t. Add *unit capacity* directed edges from s to vertices in L, and from vertices in R to t. Direct edges in E towards R, make them *unit capacity* as well.
   - Consider a maximum flow in the network. Take the set of edges between L and R that are sending 1 unit of flow. These edges form a maximum matching.



   For any set $S \subseteq L$, let $N(S) = \{v \in R \mid \exists u \in S, (u, v) \in E\}$ be the "neighbors" of S. Hall's Marriage theorem says: the size of the maximum matching in G equals $|L|$ *if and only if* for each subset $S \subseteq L$, $|N(S)| \geq |S|$. Prove Hall's theorem using the above reduction and the max-flow min-cut theorem.