Homework 4 – Due June 5th, 2017 on Canvas at 1:20pm Pacific time

Homework Guidelines

Please make sure you read the collaboration policy, and write the following as the first line of your homework: "I have read and agree to the collaboration policy. ⟨`Your name`⟩." Your homework will not be graded if you do not write this.

Collaboration policy Collaboration on homework problems is permitted, but not encouraged. You are allowed to collaborate with *at most two students enrolled in the class*. You must mention the name of your collaborators clearly on the first page of your submission. Even if you collaborate, you are *expected to write and submit your own solution independent of others*, and your collaboration should be restricted to discussions only. Also, you should be able to explain your solution verbally to the course staff if required to do so. *Collaborating with any one not enrolled in the class, or taking help from any online resources for the homework problems is strictly forbidden.*

The Computer Science Department of UCSC has a zero tolerance policy for any incident of academic dishonesty. If cheating occurs, consequences within the context of the course may range from getting zero on a particular assignment, to failing the course. In addition, every case of academic dishonesty will be referred to the student's college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course and suspension or dismissal from the university.

How to submit your solutions Each problem must be **typed** up separately (in at least an 11-point font) and submitted in the appropriate assignment box on the Canvas website as a PDF file.

This means that you will submit 3 separate files on Canvas, one for each problem!

You are strongly encouraged, but not required, to format your problem solutions in LaTeX. Template HW files and other LaTeXresources are posted on the course webpage. LaTeXis a free, open-source scientific document preparation system. Most technical publications in CS are prepared using this tool.

You might want to acquire a LaTeX manual or find a good online source for LaTeX documentation.

The top of each problem should include the following:

- your name,
- the acknowledgement to the collaboration policy,
- the names of all people you worked with on the problem (see the handout "Collaboration and Honesty Policy"), indicating for each person whether you gave help, received help or worked something out together, or "Collaborators: none" if you solved the problem completely alone.

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and, if helpful, pseudocode,
2. a proof of correctness,
3. an analysis of running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an important skill. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Points might be subtracted for illegible handwriting and for solutions that are too long. Incorrect solutions will get from 0 to 30% of the grade, depending on how far they are from a working solution. Correct solutions with possibly minor flaws will get 70 to 100%, depending on the flaws and clarity of the write up.

Assigned Problems

Exercises (Do not hand in) Chapter 4: 3,4 and 6, Chapter 5:1-3.

Following are the problems to be handed in, 25 points each. Maximum score for this homework is 100 points. We will take your best four attempted problems.

1. **Node-capacitated networks** 2-page limit – your solutions should fit on two sides of 1 page)

   A zombie infestation on earth is threatening the population of all our cities. The virus originated mysteriously at a site located at $s \in V$ and a medical facility at t ∈ V is working on a cure at the moment. You have to stop the virus from spreading across a network of cities, represented by a directed graph $G = (V, E)$, between s and t because the facility at t is your only chance at saving the planet. The cost of cutting off all possible transportation out of a single city $v \in V$, which can be thought of as a node capacity of that city, is $c_v \geq 0$. You can define a flow f in this network, given that the flow though a node v is defined as $f_{in}(v)$. We say that a flow is feasible if it satisfies the usual flow-conservation constraints and also the following constraints: $f_{in}(v) \leq c_v$ for all nodes. You have to find an s-t maximum flow through this network, cutting which will solve your problem in polynomial time. Define an s-t cut for such a network, and show that the analogue of the Max-Flow Min-Cut Theorem holds true.

   To organize your answer better break it into parts as follows:

   (a) Give an algorithm for maximum flows in node-capacitated networks of this kind. Pay close attention to the argument of correctness of your algorithm.

   [*Hint:* One way to give an algorithm for this problem is to run Ford-Fulkerson on a modified version (say $G' = (V', E')$) of the original graph G, in which each vertex in V corresponds to two vertices (call them v_{in} and v_{out}) in V'. The graph G' should have an edge e' for every edge e in E along with some additional edges.]

   (b) Define an analogue of an *s-t* cut in a node-capacitated network, and define the "capacity" of your object.

(c) Explain why the max-flow min-cut theorem holds for your analogue.

If it is easier, you may want to prove correctness of your algorithm at the same time as you prove your max-flow/min-cut theorem.

2. (**Cycle cover** 2-page limit – your solutions should fit on two sides of 1 page) Your task is to give an algorithm which finds a cycle cover for a given graph or correctly reports that no cycle cover exists. An analysis of correctness along with the time and space complexity should also be included in your solution. You should base your algorithm on a reduction to bipartite matching. (A *cycle cover* of a given directed graph $G = (V, E)$ is a set of vertex-disjoint cycles that cover all the vertices.)

3. (**Flow decomposition** 2-page limit – your solutions should fit on two sides of 1 page) A flow f is **acyclic** if there are no directed cycles in the subgraph of edges with positive flow.

   (a) Prove that every flow f has at least one corresponding acyclic flow that has the same value. (In other words, for every graph, at least one maximum flow is acyclic.)

   (b) A **path flow** is a flow that gives positive values to a simple, directed path from source to sink. Prove that every acyclic flow is a finite combination of path flows.

   (c) Some flows for a directed graph are *not* a combination of path flows. Give an example of one.