CS7800: Advanced Algorithms. Fall 2017
Homework 6

Instructor: Jonathan Ullman TA: Albert Cheu

Due Friday, October 27th at 11:59pm
(Email to neu.cs7800@gmail.com)

Congratulations on finishing the midterm. Your present is a short HW for this week!

Homework Guidelines

Collaboration Policy. Collaboration on homework problems is permitted, however it will serve you well on the exams if you solve the problems by yourself. If you choose to collaborate, you may discuss the homework with at most 2 other students currently enrolled in the class. Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, indicate that on your submission. If asked, you must be able to explain your solution to the instructors.

Preparing and Submitting Solutions. You must type your solutions using LaTeX. Please use an 11-pt or larger font. Please submit both the source and PDF files using the naming conventions `lastname_hw6.tex` and `lastname_hw6.pdf`. Your name must be on the first page of the PDF.

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. You may use any facts from class in your analysis and you may use any algorithms from class as subroutines in your solution.

You should be as clear and concise as possible in your write-up of solutions. Communication of technical material is an important skill, so clarity is as important as correctness. 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 solutions that are too long.

Problem 1 (The Dinner Party, 10 pts)**.**

Several families are having a dinner party. To be more social, they would like to sit at tables so that no two members of the same family are at the same table. There are p families, and family i has $m(i)$ members. There are q tables and each table j has $t(j)$ seats. Design a polynomial-time algorithm that finds such a seating arrangement or determine that no such arrangement exists by reducing this problem to a maximum-flow computation. You may use any maximum flow algorithm as a subroutine without describing it or proving its properties.

Your solution should include the following:

(1) An explanation of how you take the number of families and their sizes, and the number of tables and their sizes, and output a flow network G.

(2) An explanation of how you use a maximum flow in G to construct a seating assignment or determine that no suitable assignment exists.

(3) An explanation of why your reduction is correct.

(4) A statement of the running time of your algorithm, including the time to solve the maximum flow problem using the subroutine of your choice.