COL 351 Aug 16, 17, 21

TUTORIAL SHEET 3

1. **(KT-Chapter 1)** Gale and Shapley published their paper on the stable marriage problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals.

   Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals. The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.) We say that an assignment of students to hospitals is stable if neither of the following situations arises.

   - **First type of instability:** There are students s and s', and a hospital h, so that (i) s is assigned to h, (ii) s' is unassigned, and (iii) h prefers s' to s.
   - **Second type of instability:** There are students s and s', and hospitals h and h', so that (i) s is assigned to h and s' is assigned to h', (ii) h prefers s' to s, and s' prefers h to h'.

   So we basically have the stable marriage problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students. Show that there is always a stable assignment of students to hospitals, and give an efficient algorithm to find one. The input size is $\theta(mn)$; ideally, you would like to find an algorithm with this running time.

2. Consider the following algorithm for finding minimum spanning tree: sort all edges in *decreasing order* of weight. Let the edges be $e_1, \ldots, e_m$. Consider edges in this order, and initialize the set T to G, the entire graph. When we consider an edge e_i, we remove it from T if T contains a cycle containing e_i; otherwise we keep e_i. Prove that the final set T will be a minimum spanning tree (assume that G is connected).

3. You want to throw a party and is deciding whom to call. You have n people to choose from, and you have made up a list of which pairs of these people know each other. You want to pick as many people as possible, subject to two constraints: at the party, each person should have at least five other people whom they know and five other people

whom they don't know. Give an efficient algorithm that takes as input the list of n people and the list of pairs who know each other and outputs the best choice of party invitees. Give the running time in terms of n.

4. Prove the following two properties of the Huffman encoding algorithm (assume that the sum of the frequencies of the characters is 1):(i) If some character occurs with frequency more than 2/5, then there is guaranteed to be a codeword of length 1, (ii) If all characters occur with frequency less than 1/3, then there is guaranteed to be no codeword of length 1.

5. Suppose you are given a text of length n^c where c is a constant and n is the number of distinct characters in the alphabet. Show that the Huffman tree for this text has height $O(\log n)$.