Second Level Algorithms

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 07

Lecture 35

Welcome to the 35th lecture of the second-level algorithm course. So, in this lecture, we will start a new topic, which is very important: the stable matching problem, okay? So, let us begin. So, what is the setup? We have a set M of n men and a set W of n women. Every man has a preference over all women, and every woman has a preference over all men, okay? So, for a concrete example, suppose we have three men: M_1, M_2, M_3 and three women: W_1, W_2, W_3.

Suppose M_1 likes W_2 most, followed by W_1, followed by W_3. This should be M_2; this should be M_3. Suppose M_2 likes W_1, followed by W_3, followed by W_2. M_3 likes W_1, followed by W_2, followed by W_3. Suppose W_1 likes M_1, followed by M_2, followed by M_3. W_2 likes M_1, followed by M_3, followed by M_2. And suppose W_3 likes M_3, followed by M_2, followed by M_1. So, this is the input to the stable matching problem. The goal is to match men and women in a stable way. This is informal. So, let us formalize it: what do we mean by matching men and women in a stable way? To do that, we will define what instability is, and a stable matching is one which is not unstable—that is the way we define it. So, to define instability, we introduce the concept of a blocking pair. We define instability of a matching M using the concept of a blocking pair. So, what is a blocking pair? A pair, say (m, w), of a man and a woman is called a blocking pair for a matching M if any of the following conditions hold.

The first condition is: both m and w are unmatched. If both m and w are unmatched, they will get matched together. Every man and woman prefers being matched to remaining unmatched. So, they will form a matching together, and hence a new matching will emerge. So, that is why the existence of a pair (m, w), where both m and w are unmatched, makes the matching unstable.

What is matching? A set of edges sharing no endpoints. That means, in this context, every man is matched with exactly one woman, and every woman is matched with exactly one man. Okay. The second condition could be that if M is unmatched,

and W prefers M, then her partner—let's denote the partner of W under the matching M as capital M W—that is, we denote this fact as a notation. We denote the preference of the woman W with this notation, and this woman W prefers M over her currently matched partner. If this happens, it is better for W to get unmatched from her partner and get matched with M. So, the existence of such a pair (M, W) also makes the matching unstable. The symmetric opposite condition is the third condition.

If W is unmatched, and M prefers W, then his partner M M that is this. The fourth condition is both M and W are matched, but both of them prefer each other over their

Partners under M—that is, the man M prefers W over his partner under M and the woman W prefers the man M over her partner under M. Okay, so this is called a blocking pair. And then, what is stable matching? A matching M is called stable if there is no blocking pair. For M, some important questions are: Does a stable matching always exist? If yes, can we compute it in polynomial time, and so on and so forth?

So, these questions we will consider. But before that, let us understand this concept of stability and blocking pair with some examples. The first observation is that if we have the same number of men and women. So, let us denote the set of men as A and the set of women as B. We will use capital M to denote a matching.

So, if we have the same number of men and women in our instance, every stable matching must be perfect—that is, every man and woman must be matched in every stable matching. Indeed, this must be the case because if it is not a perfect matching, then there exists a man and a woman who are not matched with anybody else, and the existence of one such man and one such woman amounts to a blocking pair for that matching. Okay. So, we will assume in our discussion of stable matching, the number of men is the same as the number of women. If they are unequal, all the things that we discuss here can be easily generalized. So, to convey the main message, we assume—without loss of much generality—that we have the same number of men and women. Now, let us see a concrete example to get a feel of stable matching. So, let us see an example of three men and three women. Let us assume some preferences. Okay.

And let's assume a matching. Suppose we match M_1 with W_2, M_2 with W_3, and M_3 with W_1. That is, consider the matching M where M_1 is matched with W_2. M_2 with (let us use parenthesis) M_2 is matched with W_3, and M_3 is matched with W_1. We see that it has a blocking pair, namely M_1 and W_1.

So, M is not a stable matching. Because it has a blocking pair. So, let us consider another matching. Let us use green color to denote it: let us match M_1 and W_1, M_2 with W_2, and M_3 with W3. Let's argue that this is a stable matching. How to see that?

You see, M_1 is matched with his most preferred woman. Similarly, M_2 is matched with his most preferred woman, and M_3 is matched with his most preferred woman. So, all men are matched with their most preferred women, and hence they cannot be part of any blocking pair. This can also be seen from the woman's perspective. Every man is matched with his most preferred woman. Hence, no man can be part of any blocking pair for M_1. Thus, there cannot be any blocking pair for M_1; hence, M_1 is stable.

So, we will see that every instance of a stable matching stable matching, and moreover, such a stable matching can be computed in polynomial time—that is a famous algorithm due to Gale and Shapley, part of the reason why they were awarded the Nobel Prize. But that we will see in the next lectures. In this lecture, let us see some more application scenarios of stable matching. The first application is, of course, for deciding marriage steps between men and women. You see, in all the applications, the basic framework of stable matching cannot be used directly, but it needs to be changed suitably.

But similar ideas work. For example, in this application of finding marriage partners, it is impossible for every man to have a preference ranking over all women and vice versa. So, the preferences here will be partial rankings. The second very important application is college seat allocation. So, suppose we have a set of students, and each student needs to be assigned to some college.

Students may have preferences over colleges or over the various degrees that a college offers, and every college or various streams in every college has a particular capacity or fit. So, here you see only one side—namely, students—has preference rankings over the degrees offered by various colleges. On the other hand, colleges or departments (degrees offered by various colleges) do not have preferences over the students, or in many situations, they may also want to get good students based on some ranking from an examination. Moreover, each such course has a quota. Not every course has some predefined number of seats, and those many students can be matched to that particular degree in a particular college.

Again, we see that the basic framework of stable matching cannot be used in these practical applications, which is typically the case, but similar ideas can be used to come up with algorithmic solutions for these problems. The third one—again, a problem with huge societal impact—is allocating doctors to hospitals. Again, suppose we have a set of doctors and a set of hospitals. Each hospital has a requirement for a certain number of doctors. Each doctor has preferences, but where this problem differs from college seat allocation is that doctors Some pairs of doctors may be couples, and they may prefer to be allocated to the same hospital or nearby hospitals so that they can live together. So, this part is an extra constraint that needs to be respected when allocating doctors to hospitals, which was not there in the first two applications—namely, finding marriage partners or college seat allocation.

And so on. There are many such applications of stable matching in various societal domains, and a good book is the famous green book on stable matching, which you can refer to for various other applications. So, let us stop here. In the next lecture, we will see the celebrated Gale-Shapley algorithm. Thank you.