Algorithms – handout 2

March 10, 2004

We shall consider the weighted matching problem. We will show that the problem of finding a matching of maximum weight is reducible to the problem of finding a perfect matching of minimum weight, and vice versa. Then we will present an algorithm for finding perfect matchings of minimum weight in bipartite graphs. The algorithm also derives an optimal solution to the dual problem – that of maximizing the assignments of weights to the vertices with the constraints that the sum of weights of endpoints of an edge never exceeds the weight of the edge.

There are polynomial time algorithms that find maximum weight matchings in arbitrary graphs, but they will not be presented in class.

Following a question that was asked in class last week, we present a simple reduction from vertex cover showing that finding a minimum cardinality maximal matching is NP-hard. (NP-hardness for this problem also holds for bipartite graphs.)

Homework:

Let G be a complete bipartite graph with left hand side vertices $u_1, \ldots, u_n$, right hand side vertices $v_1, \ldots, v_n$, and all edges have weights (weight w_{ij} for edge (u_i, v_j)). A *non-crossing matching* is a matching that does not include two edges that cross each other, namely, two edges such as (u_i, v_j) and (u_k, v_l) with $i < k$ and $j > l$. Present a polynomial time algorithm that finds a maximum weight noncrossing matching in bipartite graphs.

Remark: Answering this homework assignment does not require the use of algorithmic ideas presented in the lectures about matching. The algorithm I have in mind is based on *dynamic programming*, and the purpose of this homework is to make sure that students are familier with this concept. Almost every basic textbook about algorithms contains a chapter about dynamic programming, and you may consult such a textbook if you find this homework difficult.