FALL 1998: **COMP 7/8713: Advanced Algorithms**

Due Oct 14, in class

Problems

10. (**R**) Redo the seminar scheduling problem with a different objective function. Instead of maximizing the number of seminars, design an algorithm that maximizes the utilization (total time for which the seminar room will be used).
11. (**R**) Exercise 17.3-2, page 344.
12. (**Ex**) Exercise 23-1, page 495.
13. (**Ex**) Exercise 23-2, page 495.
14. (**R**) Assuming that a given graph is represented using the adjacency list representation, design a linear-time ($O(m+n)$) algorithm to sort each of the n adjacency lists. Design an $O(n^2)$ algorithm for partial credit. Hint: Use radix sort.
15. (**R**) Assuming that a given directed graph is represented using the adjacency list representation. For each edge $e = (u, v)$, we refer to the edge $e' = (v, u)$ as it partner edge, if it exists. In general, for a given edge e, it takes $O(n)$ time to locate its partner eedge, unless we can store for each edge a pointer to its partner edge. Design a linear-time algorithm so that each edge also maintains a pointer to its partner entry (make it NIL if it does not exist).
16. (**Ex**) The course homepage contains a sketch of a solution for checking in linear time whether an undirected graph has an odd cycle. Prove all the claims on that solution sketch. Modify it so that it works for undirected graphs that may not be connected. Modify it so that it works for directed graphs, i.e., it should check whether a directed graph has a **directed** cycle.