Problem Set #4
IE 495
Due October 6, 2000

1. Write pseudo−code for an implementation of a doubly linked list. The supported operations should include

   – make_list()
   – insert()
   – delete()
   – concatenate()
   – split()

   Specify the arguments for each operation and the complexity of each operation. Your implementation should work when deleting the first and/or last item and when the list is empty.

2. Let G=(V, E) be a given directed acyclic graph. We can obtain a partial order on the set of vertices by defining

   $$v < w \Leftrightarrow \text{there is a path in } G \text{ from } v \text{ to } w$$

   Such an ordering is useful in many different settings. For instance, it can be used for analyzing precedence constraints in scheduling applications or data dependencies in parallel algorithm design. Extending this partial order to a total order is known as *topological sort*. In topological sort, our goal to define a complete ordering of the vertices such that v comes before w in the ordering if and only if $v < w$. Develop an $O(|E| + |V|)$ algorithm to perform topological sort on the vertices of a graph.

3. (Optional) The Towers of Hanoi is a puzzle consisting of three pegs A, B, and C, and n circles of varying sizes. Initially, the circles are stacked on peg A in order of decreasing size, largest on the bottom. The solve the puzzle, you must move the circles from peg A to peg C one at a time in such a way that no circle is ever placed on top of a smaller circle. Peg B may be used for temporary storage of circles.

   – Develop a recursive algorithm to solve the puzzle.
   – What data structures would you need to implement the algorithm?
   – What is the complexity of your algorithm?

4. Write a program to implement parallel quick sort using PVM. Your program should read a list of integers from a file (one integer per line) and write out the sorted list to another file (same format). The number of integers should be specified on the first line of the input file.

   – What speedup can achieve? How does this compare to the theoretical speedup?
   – How does the theoretical running time compare to the real running time (i.e. look at growth rate as the size of the input file increases).

5. Modify your program from the last assignment to perform a breadth−first traversal of the ten nearest neighbor graph starting from a specified root node and report the distance from the root to each other node in the graph.

   - Discuss the complexity of your algorithm and what data structures you used.
   - Show that your algorithm produces a shortest path from the root node to all other nodes in the graph.