CSE/EE 373 Autumn 1994
Data Structures and Algorithms
Assignment # 7 Common Mistake Solutions

1. Problem 7.16 in the text.

   Answer:

   The original input is –
   3, 1, 3, 1, 5, 9, 2, 6, 5, 3, 5.
   After sorting the first, middle, and last elements, we have –
   3, 1, 4, 1, 5, 5, 2, 6, 5, 3, 9.
   Thus the pivot is 5. Hiding it gies –
   3, 1, 4, 1, 5, 3, 2, 6, 5, 5, 9.
   The first swap is between two fives. The next swap has i and j crossing (i=6, j=2).
   Thus the pivot is swapped back with i –
   3, 1, 4, 4, 5, 3, 2, 5, 5, 6, 9.
   We now recursively quicksort the first eight elements –
   3, 1, 4, 1, 5, 3, 2, 5.
   Sorting the first three appropriate elements gives –
   1, 1, 4, 3, 5, 3, 2, 5.
   Thus the pivot is 3, which gets hidden –
   1, 1, 4, 2, 5, 3, 3, 5.
   The first swap is between 4 and 3 –
   1, 1, 3, 2, 5, 4, 3, 5.
   The next swap crosses pointers, so is undone; i points at 5, and so the pivot is swapped –
   1, 1, 3, 2, 3, 4, 5, 5.
   A recursive call is now made to sort the first four elements. The pivot is 1, and the partition does not make any changes. The recursive calls are made, but the subfiles are below the cutoff so nothing is doen. Likewise, the last three elements constitute a base case, so nothing is done. We return to the original call, which now calls quicksort recursively on the right hand side, but again, there are only three elements, so nothing is done. the result is –
   1, 1, 3, 2, 3, 4, 5, 5, 5, 6, 9,
   which is cleaned up by insertion sort.

2. Problem 7.21 from the text.

   Answer:

   The strategy used here is to force the worst possible pivot at each stage. This doesn't necessarily give the maximum amount of work (since there are few exchanges, just lots of comparisons), but it does give $\Omega(n^2)$ comaprisons. By working backwards, we can arrive at the following permutation –
   20, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 10, 2, 12, 6, 14, 1, 16, 8, 18.

A method to extend this to large numbers, when n is even is as follows: The first element is n, the middle is n-1, and the last is n-2. Odd numbers (except 1) are written in starting to the left of center in decreasing order. Even numbers are written in decreasing order by starting at the right most spot, and always skipping one available empty slot, and wrapping around when the center is reached. This method takes O(nlogn) time to generate the permutation, but is suitable for a hand calculation. By inverting the cations of quicksort, it is possible to generate the permutation in linear time.

3. Problem 7.24 from the text.

   Answer:

   Insertion sort and merge sort are stable.