CSc 220: Algorithms
Homework 3
Due in Class on Thursday September 28

Return the homework written on sheet(s) of paper with your name and CSc220 written at the top of each sheet. Please staple multiple sheets together. Remember that collaboration is allowed, but that you must write the solution on your own. Also you must acknowledge all collaborators and all sources (other than the textbook) in your solutions. Each problem is worth 10 points.

Problem 1: On input an array A of n elements, each of which is an integer in $[0..n^2]$, describe a simple method for sorting A in $O(n)$ time.

Hint: think of alternative ways of viewing the elements.

Problem 2: As we will see later in the class, a binary search tree (BST) is a data structure organized as a binary tree where each node x holds a value $key[x]$. The important property of a BST is that for any subtree of a BST rooted at x, we have that $key[x]$ is larger than the keys of all the nodes stored on the left subtree of x, and smaller than the keys of all the nodes stored on the right subtree of x. An interesting consequence of this is that by performing an "inorder" walk of the tree it is possible to output the keys stored in the tree in sorted order. An inorder walk of the tree rooted at x is the following recursive procedure

```
INORDER(x)
   IF x ≠ NIL;
      INORDER(Left[x]);
      PRINT key(x);
     INORDER(Right[x])
```

Notice that INORDER makes no comparisons since it only reads the elements in the tree in a specific order.

Given n elements can you build a BST containing the elements using $O(n)$ comparisons? If yes show your algorithm. If no, explain why you think it is impossible.

Problem 3: Given to arrays A and B each of them containing n elements, *already in sorted order*. Describe a $O(\log n)$ algorithm that finds the median of all $2n$ elements in the arrays.