Assignment #2:

Programming part:

In class we discussed the problem of finding the largest and smallest elements in an unordered array `a[0..n-1]`. The solution conceptually built up a binary tree where the leaves were the array entries and the internal nodes were pairs storing the minimum and maximum values of the leaves in the subtree rooted at that internal node. In resources there is a Pascal iterative implementation of the algorithm. Your job is to write two recursive versions of the algorithm as specified below. You are to turn in a class MinMax{} that contains two public static methods `Pair mmA( int lb, int ub, int[] a )` and `Pair mmB( int lb, int ub, int[] a )`. They each return the min and the max of the subarray `a[lb..ub]` as a `Pair` where

```
class Pair {
    int alpha;   // the smaller one
    int omega; // the bigger one
    Pair ( int a, int o ) { alpha = a; omega = o; }
}
```

Both methods should be recursive. Method `mmA` is to use the usual divide-and-conquer approach, with a $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ split. Surprisingly, such a split will not give $\lceil 3n/2 - 2 \rceil$ for all values of `n`. Analyze what is giving rise to the larger than expected number of comparisons, and by using a more refined split (which could have several cases) come up with a modified version of `mmA`, called `mmB`, that does indeed use exactly $\lceil 3n/2 - 2 \rceil$ comparisons.

Written part:

These first problems are related to the min-max problem.

- In the style of the sorting hand-out, write out the tree of posets for the min-max problem when $n = 4$.
- Determine the smallest value of n for which `mmA` and `mmB` use a different number of comparisons.
- Write down a recurrence relation $C(n)$ for the number of comparisons used by `mmB`. It can have several cases. Here's the recurrence relation for `mmA`:

$$C(n) = \begin{cases} 0 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ 2 + C(\lfloor n/2 \rfloor) + C(\lceil n/2 \rceil) & \text{if } n > 2 \end{cases}$$

- Will not be marked but a good exercise nevertheless: Prove that your recurrence relation for `mmB` satisfies $C(n) = \lceil 3n/2 - 2 \rceil$ by mathematical induction.

Additional written part:

An inversion in a sequence is an out-of-order pair; i.e., $i < j$ but $a_i > a_j$. Inversions are discussed briefly in the book on page 252. For example, the sequence $(5, 3, 2, 1, 4)$ has 7 inversions.

- What is the minimum number of inversions of a permutation of $1, 2, \ldots, n$?
- What is the maximum number of inversions of a permutation of $1, 2, \ldots, n$?
- Explain carefully how to use red-black trees to compute the number of inversions in a permutation time $O(n \log n)$. Effectively, you may need to modify the code for Algorithm 3.4 on page 439. Explain in detail any changes that you would make to `put`.