GENERAL INSTRUCTIONS: Attempt all questions; do not spend too much time on any one — the point distribution is marked clearly — budget your time accordingly. No penalties will be assessed for wrong answers. Keep your work neat and clearly indicate your answers. You may use the back of the sheets if you need additional space.

(1) We will concentrate on selection and sorting in this group of questions:

   a. Briefly define the selection problem. (5 pts.)

   b. Briefly define the sorting problem. (5 pts.)

   c. Compare the relative complexities of selecting and sorting by supplying upper bounds (in terms of the "O" notation) and lower bounds (in terms of the "Ω" notation) on the two problems. Explain your answers. (10 pts.)
d. Suppose our model of computation did not incur "charges" for sorting, \textit{i.e.}, we can sort for free. Discuss how the complexity of selection (\textit{e.g.}, finding the max) is affected (element-to-element comparisons are still charged one time unit though). (10 pts.)

e. Suppose instead that our model of computation did not incur "charges" for selection, \textit{e.g.}, we can select the largest element for free. Discuss how the complexity of sorting is affected (again, element-to-element comparison still incur one unit time). (10 pts.)
(cont.)

f. Are the assumptions regarding the models of computation in (d) and (e) reasonable? Why or why not? (5 pts.)

(2) Let us relate searching and sorting with special type of graph -- trees. A binary search tree (or BST) is a binary tree whose nodes contain orderable objects which satisfy the following conditions at all "levels" in the tree:

The object $x$ at a given node is larger than all objects $y$ in that node's left subtree and is smaller than all objects $z$ in that node's right subtree.

Let us exhibit all the possible BSTs for the set $\{1,2,3\}$:

The “shape” of a BST depends crucially on the order in which its elements were added to the tree. For example, if the elements came sequentially from the list $(1,2,3)$, the BST would grow as follows:
And if the elements came sequentially from the list (2,3,1), the BST would grow as follows:

```
  2
 --+--
  3 1
```

It would seem then that different lists spawn different BSTs.

a. There are 3!=6 different lists that can be made from the set {1,2,3}. Explain why there are only 5 distinct BSTs for this set. (5 pts.)

b. To see if a particular item is in a BST, we can run the following recursive Boolean function:

```java
boolean isIn( x: item; t: BST ) {
    if ( x == t.root() )
        return true;
    else if ( x > t.root() )
        return isIn( x, t.leftSubtree() );
    else
        return isIn( x, t.rightSubtree() );
}
```

Explain why the following recurrence describes the worst case complexity of the Boolean function `isIn` (here, \( n \) is the number of items in the BST):

\[
t(n) = t(n-1) + 2
\]

which means that \( t(n) \) is \( O(n) \). (10 pts.)
c. Explain why the following recurrence describes the best case complexity of $\text{isIn}$:

$$t(n) = t(n/2) + 2$$

which means that $t(n) = \Omega(\lg n)$. (10 pts.)

d. Describe how you might extract the maximum element from a BST. (10 pts.)
(3) Compute the minimum-spanning-tree of the following graph:  (10 pts.)

4. Propose a graph-based solution to the following problem (you don't have to solve the problem itself; the solution scheme is what we’re after here):

   Among people, a celebrity is someone who everyone knows but who knows no one. To identify a celebrity, if one exists, you are allowed to ask questions of any of the people, but only of the form: "Excuse me, do you know that person over there?" Assume that all answers given are correct. Minimize the number of questions you need to ask to determine the celebrity, if one exists, or to determine that no celebrity exists in a given set of $n$ people.  (10 pts.)