CS624: Analysis of Algorithms

Assignment 3

Due: Monday, Oct. 20 2025

Topics: Sorting Bounds, Medians and order statistics, BST.

1. Given $n > 2$ distinct numbers, what is the minimum number of comparisons needed to find a number that is neither the maximum nor the minimum?
2. Argue that `Randomized-Select` never makes a recursive call to an empty array.
3. Look at the following version of `RANDOMIZED-SELECT`:

Algorithm 1 RandomizedSelect2(A,p,r,i)

if $p = r$ **then**
    return $A[p]$
end if
$q \leftarrow RandomizedPartition(A, p, r)$
$k \leftarrow q - p + 1$
if $i \leq k$ **then**
    return $RandomizedSelect2(A, p, q, i)$
else
    return $RandomizedSelect2(A, q + 1, r, i - k)$
end if

Argue that in the worst case, `RandomizedSelect2` never terminates.

4. Exercise 1.1 in Lecture note 7 (binary search trees).
5. Exercise 1.2 in Lecture note 7 (binary search trees).
6. Exercise 1.3 in Lecture note 7 (binary search trees).
7. Exercise 1.4 in Lecture note 7 (binary search trees).
8. Exercise 1.5 in Lecture note 7 (binary search trees).
9. Exercise 1.6 in Lecture note 7 (binary search trees).
10. Exercise 2.2 in Lecture note 7 (binary search trees).
11. Exercise 4.1 in the Lecture 7 handout - you have to prove all these properties are correct (for many of the proofs - use contradiction) Please be careful. This is an exercise about binary search trees, not about algorithms used to construct those trees. So all you can use in doing this problem is the definition of a binary search tree. If you write something like, "this can't happen because the algorithm would have placed this element somewhere else", then your reasoning can't possibly be correct.