Lab 5

1. Show all steps of In-Place QuickSort in sorting the array [1, 6, 2, 4, 3, 5] when doing first partition. Use leftmost values as pivots.

2. In our average case analysis of QuickSort, we defined a *good self-call* to be one in which the pivot x is chosen so that number of elements < x is less than 3n/4, and also the number of elements > x is less than 3n/4. We call an x with these properties a *good pivot.* When n is a power of 2, it is not hard to see that at least half of the elements in an n-element array could be used as a good pivot (exactly half if there are no duplicates). For this exercise, you will verify this property for the array A = [5, 1, 4, 3, 6, 2, 7, 1, 3] (here, n = 9). Note: For this analysis, use the version of QuickSort in which partitioning produces 3 subsequences *L, E, R* of the input sequence *S.*
   a. Which x in A are good pivots? In other words, which values x in A satisfy:
      i. the number of elements < x is less than 3n/4, and also
      ii. the number of elements > x is less than 3n/4
   b. Is it true that at least half the elements of A are good pivots?

3. *Interview Question.* Give an o(n) ("little-oh") algorithm for determining whether a sorted array A of distinct integers contains an element m for which A[m] = m. You must also provide a proof that your algorithm runs in o(n) time.

4. Prove the following recursive factorial algorithm is correct.

   Algorithm recursiveFactorial(n)
   Input: A non-negative integer n
   Output: n!
   if (n = 0 || n = 1) **then**
   return 1
   return n * recursiveFactorial(n-1)

5. Review of SubsetSum Problem: Given a set S = {s_0, s_1,s_2, …, s_{n-1}} of positive integers and a non-negative integer k, find a subset T of S so that the sum of the integers in T equals k or indicate no such subset can be found.

   We have already seen a brute force solution to this problem in an earlier lab. In this exercise, you are going to come up with a recursive solution for SubsetSum. Write the pseudo code for your algorithm.

   Hint:

We are seeking a $T \subseteq S = \{s_0, s_1, \ldots, s_{n-2}, s_{n-1}\}$ whose sum is k. Such a T can be found if and only if one of the following is true:

(1) A subset T_1 of $\{s_0, s_1, \ldots, s_{n-2}\}$ can be found whose sum is k, OR

(2) A subset T_2 of $\{s_0, s_1, \ldots, s_{n-2}\}$ can be found whose sum is $k - s_{n-1}$

If (1) holds, then the desired set T is T_1. If (2) holds, the desired set T is $T_2 \cup \{s_{n-1}\}$.