# Advanced Analysis of Algorithms - Quiz I

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

## 1 Instructions

1. Attempt as many problems as you can. You will be given partial credit.
2. The duration of this quiz is 1 hour, 10 minutes, i.e., 8 : 00-9 : 10 am.
3. Each question is worth 4 points for a total of 20 points.

## 2 Problems

1. We discussed the *Hat Paradox* problem in class. To recap, $N$ men throw their hats in a ring and after a random shuffling, each man draws out a hat. We showed that the expected number of people who get their own hats is 1, irrespective of the number of people involved. Derive an upper bound on the probability that all the men involved get their own hats?
2. Consider the Towers of Hanoi problem. There are 3 pegs, viz., $P_1$, $P_2$ and $P_3$. On $P_1$ there is collection of $N$ disks, such that the circumference of each disk is larger than the disk above it. The goal is to move the $N$ disks from $P_1$ onto $P_2$, subject to the following constraints:
   (a) In one move, you can move precisely one disk from one peg to another peg.
   (b) At no time, can a disk be placed on another disk with a smaller circumference.

   Argue with an example, that the additional peg, $P_3$, is *necessary*. Derive a recurrence relation for calculating the number of moves required to move $N$ disks from $P_1$ onto $P_2$. Solve the recurrence, i.e., obtain a closed-form expression for the recurrence derived.
3. Let $\mathbf{A}$ denote an array of $n$ elements. Describe a strategy that finds the maximum and minimum of the array, using at most $\frac{3}{2}n$ comparisons.
4. Show that for any integer $n \geq 0$,
$$\sum_{k=0}^{n} \binom{n}{k} \cdot k = n \cdot 2^{n-1}.$$
   *Hint: What is* $\sum_{k=0}^{n} \binom{n}{k}$ *?*

5. Professor Smith proposes a new sorting algorithm called *Smith-Sort*

**Function** SMITH-SORT($\mathbf{A}, i, j$)
**if** $(A[i] > A[j])$ **then**
    Exchange these elements in the array.
**end if**
**if** $((i+1) \geq j)$ **then**
    **return**
**end if**
$k \leftarrow \lfloor \frac{j-i+1}{3} \rfloor$
SMITH-SORT($\mathbf{A}, i, j-k$)
SMITH-SORT($\mathbf{A}, i+k, j$)
SMITH-SORT($\mathbf{A}, i, j-k$)

**Algorithm 2.1:** SMITH-SORT

Prove that Algorithm (2.1) correctly sorts an array of $n$ elements provided in $\mathbf{A}[i]$ through $\mathbf{A}[j]$. You may assume that all elements are distinct.