CSE 21 Fall 2022
Homework 3 problems, Due Oct. 13, 11:59 PM

Instructions
Students should feel free to discuss these problems.
Key Concepts Recursive algorithms, recurrence relations.

Answer one of the two review problems. The grade for the review problems will also add to the corresponding previous homework grade, up to 80% . Then answer all of the other problems.

1. (Review for homework 1)

   I am on a committee with n people which has many subcommittees. Each sub-committee has exactly three members from my committee. Let SC_i be the number of subcommitees the i'th committee member is on. Prove that $\sum_{i=1}^{i=n} SC_i$ is divisible by 3. (20 points)

2. (Review for homework 2)

   Consider the following algorithm to take a binary integer $b_{n-1}..b_0$ representing the integer $\sum_{i=0}^{i=n-1} b_i 2^i$ to decimal. It uses a sub-procedure *Add* that adds two decimal numbers.

   ```
   Convert[$b_{n-1}..b_0$]
   (a) $X = b_{n-1}$
   (b) For $i = n-2$ down to 0 do
           $X = Add(X, X)$
           $X = Add(X, b_i)$
   (c) Return $X$.
   ```

   Prove that this algorithm returns the integer whose binary representation is the input using an appropriate loop invariant. (10 points) Give a time analysis assuming *Add* is constant time (5 points). Then give a time analysis assuming that *Add* takes linear time in terms of the number of decimal digits in the two inputs. (You will first have to think about how many digits is the decimal representation on an integer that takes n bits in binary.) (5 points)

3. Solve the following recurrence relations to give closed form expressions of the functions (4 points each, 2 points correct answer, 2 points explanation):

   (a) $F[0] = 1$, $F[n] = 3F[n-1]$ for $n \geq 1$.
   (b) $G[0] = 1$, $G[n] = G[n-1] + n$ for $n \geq 1$
   (c) $H[0] = H[1] = H[2] = 1$, $H[n] = 2H[n-3]$ for $n \geq 3$.
   (d) $J[1] = 1, J[n] = (n)(n-1)J[n-1]$ for $n \geq 1$.
   (e) $K[0] = K[1] = 1, K[n] = 4K[\lfloor n/2 \rfloor]$ for $n \geq 2$.

4. Give a recursive description of binary search. (10 points)

   Give a recurrence relation for its time , and solve the recurrence relation to give the time up to order. Assume that when we call a procedure on a sub-array, we do not have to copy the whole sub-array, but can just pass on the end markers to the recursive call. (10 points)

5. Consider how lines split the plane into regions. For example, two intersecting lines split the plane into four regions. Give a recurrence for $R(n)$, the maximum number of possible regions defined by n lines. (10 points, explain your answer) Solve this recurrence to get a closed form expression. (10 points)

6. Present an algorithm that, given an array $A[1..n]$ of integers (some of which might be negative), finds the consecutive sub-sequence $A[I..J]$ that has as large a sum $\sum_{i=I}^{i=J} A[i]$ as possible. For example, if $A[1..5] = (3, -1, 2, -3, 2)$, the largest sum is $A[1.3]$ which has total $3 - 1 + 2 = 4$. Prove that your algorithm is correct. Make your algorithm as fast as you can, and give a time analysis. (5 points clear description, 5 points proof of correctness, 5 points if $O(n \log n)$ or better, and 5 points for clear time analysis.)