MATH182 MIDTERM
DUE July 7, 2020

Question 1. *Consider the following pseudo-code:*

```
sum = A[1]
max = sum
for j = 2 to A.length
    sum = sum + A[j]
    if sum > max
        max = sum
return max
```

This algorithm takes as input an array $A[1\,..\,n]$ and outputs the value of the maximum subarray of the form $A[1\,..\,j]$, i.e., it outputs the number

$$\max\left\{\sum_{i=1}^{j} A[i] : 1 \leq j \leq A.\mathit{length}\right\}$$

*(1) Give a proof of the correctness of this algorithm. Your proof should include: a precise statement of a loop invariant for the **for** loop, and a proof of this loop invariant. (5pts)*

(2) Analyze the running-time of this algorithm. This includes deducing a tight asymptotic bound. (5pts)

(3) Is this algorithm asymptotically optimal (i.e., is there another algorithm with asymptotically smaller running time which can do the same thing this algorithm does)? Justify your answer. (2pts)

Question 2. *Recall that for $0 \leq k \leq n$, the **binomial coefficent** $\binom{n}{k}$ is defined by*

$$\binom{n}{k} := \frac{n!}{k!(n-k)!}$$

In particular, we have $\binom{n}{0} = \binom{n}{n} = 1$ for every n.

(1) Prove for every $0 < k < n$:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

You can use any valid method you know to prove this (from the definition, combinatorial, generating function, etc.) (5pts)

(2) Write pseudocode for a <u>recursive</u> algorithm Binomial(n, k) *which returns $\binom{n}{k}$. Your algorithm should use the above fact you proved in (1). (5pts)*

(3) Give a proof of correctness of your algorithm in (2). You should prove the statement: "For every $n \geq 0$ and for every $0 \leq k \leq n$, Binomial(n, k) *returns $\binom{n}{k}$." (5pts)*

Question 3. *Use a recursion tree and the substitution method to guess and verify an asymptotically tight bound for the following recurrence (5pts):*

$$T(n) = 2T(n-1) + 1$$

Question 4. *For the following recurrence determine an asymptotically tight bound using any method (recursion tree and substitution, master method, etc.). (5pts)*

$$T(n) \;=\; 25T(n/5) + \frac{n^2}{\lg n}$$

Question 5. *For the following functions $f(n)$ and $g(n)$, determine whether they satisfy:*

(1) $f(n) = o(g(n))$,
(2) $f(n) = \Theta(g(n))$, *or*
(3) $f(n) = \omega(g_n)$.

The functions are:

$$f(n) \;=\; (\lg n)^{\sqrt{\lg n}} \quad \text{and} \quad g(n) \;=\; \sqrt{n}$$

Justify your answer. (5pts)

Question 6. *(True/False) For each of the following statements indicate whether they are* ***true*** *or* ***false***. *Each question is worth 2pts, a blank answer will receive 1pt. Recall that "true" means "always true" and "false" means "there exists a counterexample".*

(1) *For every* $n \geq 1$ *and* $a, b \in \mathbb{Z}$, *if* $ab \bmod n = 0$, *then either* $a \bmod n = 0$ *or* $b \bmod n = 0$.
(2) *Let* $(F_n)_{n \geq 0}$ *be the sequence of Fibonacci numbers, so* $F_0 = 0, F_1 = 1$ *and for every* $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. *Then for every* $n \geq 2$, $F_{2n} = F_{2(n-1)} + F_{2(n-2)}$.
(3) *Suppose* $f(n)$ *and* $g(n)$ *are asymptotically positive, polynomially bounded functions. If* $f(n) = \Theta(g(n))$, *then* $2^{2^{f(n)}} = \Theta(2^{2^{g(n)}})$.
(4) $\Omega(n) = O(n^2)$.
(5) *The best-case running time of* Insertion-Sort *is* $O(n \lg n)$.
(6) Merge-Sort *is an asymptotically optimal comparison-based sorting algorithm.*