Assignment 1 (6+5+4+5 points):

a) Specify (!) and describe three significantly different algorithms for sorting $n$ given keys, together with their asymptotic computational worst-case costs. (No proofs required.)

b) Set up and justify a recurrence for the number $T(n)$ of steps performed by the recursive sorting algorithm $\text{stoogeSort}$ shown below when called with $\text{left}=0$ and $\text{right}=n-1$.

```c
procedure stoogeSort(int array[], int left, int right)
    if array[left] > array[right] then swap(array[left], array[right]) fi
    if (right - left + 1) < 3 then return fi
    int third := (right - left + 1) / 3 // rounding down
    stoogeSort(array, left, right-third)
    stoogeSort(array, left+third, right)
    stoogeSort(array, left, right-third)
endproc
```

c) Let $\text{array[]} = (3, 6, 5, 2, 1, 4)$, initially, and consider the call $\text{stoogeSort\_array(0, 5)}$. Write the contents of the array and variables after execution of (i) line #4, (ii) line #5, (iii) line #6, and (iv) line #7. (Do not expand the recursive calls themselves, though; instead peruse the fact that $\text{stoogeSort}$ correctly sorts arrays of size up to 4...)

d) Prove that the asymptotic growth of any non-decreasing $f: [1; +\infty) \rightarrow [1; +\infty)$ satisfying $f(n)=b\cdot f(n/a)$ for all $n \geq a$ is $f(n)=\Theta(n^{\log_b(a)})$, if $1 < a < b$ are fixed.

Reminder from calculus: $a^x = e^{x \ln(a)}$, $\log_a(y) = \ln(a)/\ln(y)$, $\ln(3)/\ln(1.5) \approx 2.71$

Assignment 2 (5+5 points):

a) Explain in few sentences, and give examples demonstrating, the differences between (i) a program/code, (ii) an algorithm, and (iii) a heuristic.

b) Briefly explain the difference between (i) cost of an algorithm and (ii) computational complexity of a problem, for instance by comparing Assignments 1a) and 1b).
Problem 3 (4+5+6+4+6 points):

a) What is the worst-case cost (=number of bit flips) of once incrementing a binary counter containing any integer between 0 and \( n-1 \), asymptotically as \( n \to \infty \)? Justify your answer by (i) exhibiting an example where that many bit flips do occur and (ii) by proving that more bit flips cannot occur.

b) Analyze the amortized cost of a binary counter with operation \( \text{INC} \).
   That is, when counting in binary from 0 to \( n \), determine the total number of bit flips, divided by \( n \) asymptotically as \( n \to \infty \). Again, justify your answer!

c) Now analyze the amortized cost of a binary counter with both operations \( \text{INC} \) and \( \text{DEC} \).
   That is, determine the total number of bit flips, divided by \( n \), incurred in the worst case by any combination of \( n \) calls to \( \text{INC} \) and/or \( \text{DEC} \) asymptotically as \( n \to \infty \). Justify!
   (Initially the counter contains zero, and decrementing zero returns zero again...)

d) Analyze the worst-case cost of the following algorithm, asymptotically as \( n \to \infty \).
   Given an \( n \)-tuple \( (b_1...b_n) \) of bits, scan for the (index \( j \) of the) first bit that is non-zero; in case all \( b_j \) are zero, count to \( 2^n-1 \) and stop.

e) Now analyze its asymptotic average cost. Justify your answers!

Problem 4 (5+5+5 points): Suppose \( \mathcal{A} \) is a randomized algorithm solving the decision problem \( L \) in time \( t(n) \) with one-sided error \( \frac{1}{2} \) independently of \( n \): On inputs \( x \notin L \), \( \mathcal{A} \) always correctly reports \( \text{false} \); but on inputs \( x \in L \), \( \mathcal{A} \) might also report \( \text{false} \) with probability \( \frac{1}{2} \).

a) Design and analyze an algorithm \( \mathcal{A}' \) that, by repeating \( \mathcal{A} \) an appropriate (which?) number \( N \) of times, errs with probability \( \leq 2^{-100n} \), where \( n \) denotes the (known) length of \( x \).

b) Analyze the (i) worst and (ii) expected cost of the following randomized 'algorithm':
   Flip a fair coin. If it comes out heads, stop; otherwise repeat.

c) Prove \( \sum_n n \cdot p^n = p/(1-p)^2 \) for all \( |p|<1 \). Hint: Cancel one \( p \) and compare anti-derivatives.