Practice Exam 1 Solutions

You will have 1 hour for this exam, although you should not need that long. This exam is open book and note. Please take some time to check your work. If you need extra space, write on the back. You must show your work to receive any partial credit. There are a total of 30 points on this exam.

1. (6 points) Consider the following Scala function:

```
def m(a: Int, b: Int): (Int, Int) = {
  var x = a
  var y = 0
  while (x >= b) {
    x = x - b
    y = y + 1
  }
  (y, x)  // Return this pair
}
```

x	y
10	0
7	1
4	2
1	3

(a) What is the result of `m(10, 3)`?

(3, 1)

(b) Give an invariant relating the values of `x` and `y` each time the `while` test is evaluated:

$a = x + b \cdot y$

(c) What function is computed by `m(a, b)`? Support your claim using your invariant. You should assume that $a \geq 0$ and $b > 0$.

$m(a, b) = (a / b, a \% b)$

when loop exits, know $a = x + b \cdot y$ (inv.)

and also $x < b$

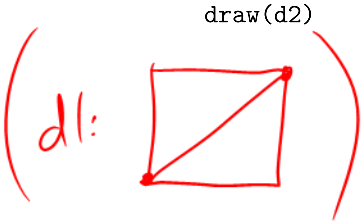
$\Longrightarrow$ y is a/b (int. division)

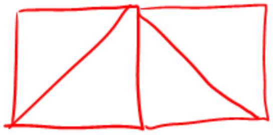
and x is $a \% b$ (remainder)

2. (6 points) This question deals with our Turtle Drawing Language.

(a) What picture will be produced from the following:

```
val d1 = Overlay(Square(100), Polygon(List((0, 0), (100, 100))))
val d2 = Overlay(d1, Offset(Rotate(d1, 90), 200, 0))
draw(d2)
```





(b) Complete this function to count the number of line segments in a drawing:

```
def lines(d: Drawing): Int = d match {
    case Square(size) => 4
    case Circle(size) => 360
    case Polygon(points) => {
        def aux(pts: List[(Double, Double)]): Int = pts match {

            case Nil => 0

            case head :: Nil => 0

            case head :: tail => 1+ aux(tail)
        }
        aux(points)
    }
    case Nothing => 0

    case Overlay(d1, d2) => lines(d1)+ lines(d2)

    case Offset(d1, x, y) => lines(d1)

    case Rotate(d1, a) => lines(d1)

    case Scale(d1, f) => lines(d1)

    case PenColor(d1, c) => lines(d1)

    case FillColor(d1, c) => lines(d1)
}
```

(c) Given the above definitions, what should be the result of `lines(d2)`?

10

3. (6 points) Suppose the running time $T(N)$ of some algorithm is given by the following recurrence:

$$\begin{cases} T(1) = 1 \\ T(N) = T(N-1) + 2N - 1, & (N > 1) \end{cases}$$

(a) Fill in the following table of values. For the last entry, give a closed-form expression for $T(N)$, either by solving the recurrence or by guessing:

$T(1)$	$T(2)$	$T(3)$	$T(4)$	$T(N)$
1	4	9	16	N^2

solving the recurence:

$$T(N) = T(N-1) + 2N - 1$$
$$= (T(N-2) + 2(N-1) - 1) + 2N - 1$$
$$= T(1) + 2\cdot(2 + 3 + \ldots + N) - (N-1)$$
$$= 2\cdot(1 + 2 + 3 + \ldots + N) - N = 2\cdot\frac{N(N+1)}{2} - N = N^2.$$

(b) Prove by induction that your closed-form expression for $T(N)$ is correct.

base case: $T(1) = 1 = 1^2$ ✓

ind. step: assume $T(N) = N^2$ for some $N \geq 1$

then $T(N+1) = T(N) + 2(N+1) - 1$

$= N^2 + 2N + 1$ by I.H.

$= (N+1)^2$ ✓

4. (12 points) Here is our Scala code for inserting a value in a list:

```
def insert(n: Int, nums: List[Int]): List[Int] = nums match {
  case Nil => n :: Nil
  case head :: tail =>
    if (n <= head)
      n :: nums
    else
      head :: insert(n, tail)
}
```

(a) Complete the following skeleton to define a function `insertAll` which takes two lists of numbers and returns a new list with all of the numbers from the second inserted into the first:

```
def insertAll(nums1: List[Int], nums2: List[Int]): List[Int] = nums2 match {
  case Nil =>
          nums1

  case head :: tail =>
      insert(head, insertAll(nums1, tail))      ] both
                    - or -                      | work
      insertAll(insert(head, nums1), tail)      ] OK
}
```

(b) Show the list which results from evaluating `insertAll(Nil, List(3, 1, 4, 1, 5))`:

= insert(3, insert(1, insert(4, insert(1, insert(5,Nil)))))

= List(1,1,3,4,5)

(c) Give a tight big-oh upper bound on the average running time of `insertAll` in terms of the size of the second list, N (assume that the first list is empty):

insert(n, nums) is O(nums.length),

so insertAll(Nil, nums2) is

$O(0 + 1 + 2 + \ldots + (N-1)) = O(N^2)$.

(continued)

(d) When the first list is not empty, what precondition do we need on `insertAll` to ensure that the resulting list will be ordered?

nums1 must be ordered

(e) Assuming the precondition from the previous question is met, give a tight big-oh upper bound on the average running time of `insertAll(nums1, nums2)` in terms of the sizes N_1 and N_2 of the two input lists:

$$N_1 + (N_1+1) + \dots + (N_1 + N_2 - 1)$$

$$= N_2 \cdot \left(\frac{N_1 + (N_1 + N_2 - 1)}{2} \right) = \frac{1}{2} N_2 (2N_1 + N_2 - 1)$$

$$= O(N_1 N_2 + N_2^2)$$

(f) If we know that both lists `nums1` and `nums2` are ordered, then is there a faster way to produce the same result as `insertAll(nums1, nums2)`? Name the desired operation, and give its big-oh average running time:

merge: $O(N_1 + N_2)$