CS:3330 Mini-Exam 1, Fall 2015

Due: Tue, Dec 8 2015 by class time (9:30 am)

---

1. Write down the worst case running time of each of the following code fragments as a function of n. Use the Θ-notation to express your answers. Show your work to receive partial credit.

(a) **for** $i \leftarrow 1$ **to** n **do**
   $j \leftarrow 1$
   while $j \leq n$ **do**
      `print`("`hello`")
      $j \leftarrow 3 * j$

(b) $B \leftarrow n$
for $i \leftarrow 1$ **to** n **do**
   for $j \leftarrow 1$ **to** B **do**
      `print`("`hello`")
   $B \leftarrow \lfloor 3 * B/4 \rfloor$

(c) **for** $i \leftarrow 1$ **to** n **do**

```
for i ← 1 to n do
    j ← 1
    while j ≤ n do
        print("hello")
        j ← j + 10
```

(d)

```
for i ← 1 to n do
    for j ← n downto i do
        sum ← 0
        for k ← i to j do
            sum ← sum + k
        print(sum)
```

2. Here is a randomized algorithm that attempts to determine if a given positive integer n is a prime number or a composite.

```
f ← random(2, ⌈√n⌉)
if f evenly divides n then
    return "composite"
else
    return "prime"
```

(a) Under what circumstances will this algorithm return an incorrect answer? Will the algorithm always correctly identify composite numbers? Will the algorithm always correctly identify prime numbers?

(b) The number 541 is a prime and 541*541 = 292681. Suppose we provide $n = 292681$ as input to the algorithm I show above. What is the probability that n will be classified as a composite?

(c) Call a positive integer n *good* if at least 1/10-th of the integers in the range $[2, \lceil\sqrt{n}\rceil]$ are its factors. What is the probability that the algorithm will incorrectly classify a good input as a prime?

(d) Now suppose that we want to decrease the error probability of the algorithm to at most 1/10, whenever the input is good. What changes would you make to the above algorithm to achieve this?