CSCI 3104, Algorithms **Prof. Aaron Clauset**
Problem Set 1 **Spring 2017, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

---

1. (15 pts total) For each of the following claims, determine whether they are true or false. Justify your determination. If the claim is false, state the correct asymptotic relationship as O, Θ, or Ω.
   (a) $n+3 = O(n^3)$
   (b) $3^{2n} = O(3^n)$
   (c) $n^n = \Theta(n!)$
   (d) $1/(3n) = \Omega(1)$
   (e) $\ln^3 n = \Theta(\lg^3 n)$

2. (15 pts total) Simplify the following expressions. Show your work.
   (a) $\frac{\mathrm{d}}{\mathrm{d}t}\left(2t^4 + \frac{1}{2}t^2 - 7\right)$
   (b) $\sum_{i=0}^{k} 2^i$
   (c) $\Theta(\sum_{k=1}^{n} 1/k)$

3. (5 pts) The young wizards Crabbe and Goyle are having an argument about an algorithm A. Crabbe claims, vehemently, that A has a running time of at least $O(n^2)$. Explain why Crabbe is spouting nonsense.

4. (10 pts total) Using the mathematical definition of Big-O, answer the following. Show your work.
   (a) Is $2^{n\,k} = O(2^n)$ for $k > 1$?
   (b) Is $2^{n+k} = O(2^n)$, for $k = O(1)$?

5. (15 pts) Exercise 2.1-3 in CLRS

6. (5 pts) Crabbe and Goyle are at it again. This time, Crabbe is claiming, vehemently, that when n real-valued numbers are arranged in sorted order in an input array A, it is always more efficient to use binary search to find a target v in the array. Goyle claims, loudly and just as vehemently, that sometimes linear search, i.e., one that scans from $A[1]$ to $A[n]$ in order, can be faster. Explain who is correct.

7. (5 pts) Crabbe has written some code to apply a wizard function `w()` to some numbers, stored in an input array $A[i, j]$ for $1 \leq i, j \leq n$. Assume `w()` takes $O(1)$ time to compute. Determine the asymptotic running time of Crabbe's function.

```
wizardAllTheThings(A) {
   for i = 1 to n {
      for j = i/2 to n { w(A[i,j]) }
   }
   return
}
```

8. (10 pts) Exercise 3.1-5 in CLRS

9. (20 pts total) Crabbe and Goyle are now arguing about binary search. Goyle writes the following pseudocode on the board, which he claims implements a binary search for a target value `v` within input array `A` containing n elements.

```
bSearch(A, v) {
   return binarySearch(A, 0, n, v)
}

binarySearch(A, l, r, v) {
   if l >= r then return -1
   p = floor( (l + r)/2 )
   if A[p] == v then return m
   if A[m] < v then
     return binarySearch(A, m+1, r, v)
     else return binarySearch(A, l, m-1, v)
}
```

   (a) Help Crabbe determine whether this code performs a correct binary search. If it does, prove to Goyle that the algorithm is correct. If it is not, state the bug(s), give line(s) of code that are correct, and then prove to Goyle that your fixed algorithm is correct.

   (b) Goyle tells Crabbe that binary search is efficient because, at worst, it divides the remaining problem size in half at each step. In response Crabbe claims that trinary search, which would divide the remaining array A into thirds at each step, would be *even more* efficient. Explain who is correct and why.