CSCI 3104, Algorithms **Prof. Aaron Clauset**

Problem Set 1 **Spring 2014, CU-Boulder**

Note: When a question asks you to give/describe/present an algorithm, you need to do four things to receive full credit:

1. Design the most efficient algorithm possible. Reasonable partial credit will be given for less efficient algorithms, as long as they are still correct, well-presented, and correctly analyzed.
2. Describe your algorithm succinctly, using structured English/pseudocode. We don't want compilable source code but note that plain English exposition is usually not enough. Follow the examples given in the textbook and lectures.
3. Justify the correctness of your algorithm, including its termination if that is not obvious.
4. Analyze the time and space complexity of your algorithm.

For other problems, remember that you must always justify your answer by showing how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

---

1. (3 pts each) For each of the following claims, determine whether they are true or false. Justify your determination.

   (a) $n + 3 = O(n^3)$
   (b) $3^{2n} = O(3^n)$
   (c) $n^n = o(n!)$
   (d) $1/(3n) = o(1)$
   (e) $\ln^3 n = \Theta(\lg^3 n)$

2. (4 pts each) Simplify the following expressions. Show your work.

   (a) $\frac{\mathrm{d}}{\mathrm{d}t}\left(3t^4 + \frac{1}{3}t^3 - 7\right)$
   (b) $\sum_{i=0}^{k} 2^i$
   (c) $\Theta(\sum_{k=1}^{n} 1/k)$

3. (23 pts) T is a balanced binary search tree storing n values. Describe an $O(n)$-time algorithm that takes input T and returns an array containing the same values in *ascending* order.
   Hint: this is an algorithm you have probably encountered before, so pseudocode is not necessary if you can give the correct name and explain in terms of its operations why it always produces the correct result.

4. (10 pts each) Acme Corp. has asked Professor Flitwick to develop a faster algorithm for their core business. The current algorithm runs in $f(n)$ time. (For concreteness, assume it takes $f(n)$ microseconds to solve a problem of size exactly n.) Flitwick believes he can develop a faster algorithm, which takes only $g(n)$ time, but developing it will take t days. Acme only needs to solve a problem of size n once. Should Acme pay Flitwick to develop the faster algorithm or should they stick with their current algorithm? Explain.

   (a) Let $n = 41$, $f(n) = 1.99^n$, $g(n) = n^3$ and $t = 17$ days.

   (b) Let $n = 10^6$, $f(n) = n^{2.00}$, $g(n) = n^{1.99}$ and $t = 2$ days.

   Hint: start with the *largest* n Acme can solve with Flitwick after t days of waiting.

5. (5 pts each) 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 $k = O(1)$?

6. (20 pts) Is an array that is in sorted order also a min-heap? Justify.