Participation 5

Due Thursday, July 17, at 9 AM (GMT-7)

CSc 345 – Summer 2014

Instructor: Qiyam Tung

Instructions

1. This is an individual assignment. You must do your own work.
2. If you are having difficulty and need to ask a question you can:
   (a) Ask questions in class.
   (b) Stop by my office hours (or make an appointment).
   (c) Post a question on Piazza.
   (d) Post a private question on Piazza if the question is too specific.
3. Show all work. I will be grading on whether you put effort into this problem (i.e. participation) and not correctness. Showing your work helps me identify your thought process and helps me with grading.
4. You may write your solutions by hand, or you may type them using any appropriate program such as Microsoft Word, OpenOffice Writer, LaTeX, etc. . .
   However, the final copy should be in PDF form and formatted so that it is legible.

L-Systems

L-systems are a type of formal grammar. A grammar is set of production rules for strings in a language. These sets of rules produce valid strings of a language. What the strings mean is not encoded in the grammar itself; it is only concerned with the structure and form of the strings.

Grammars have applications in theoretical computer science, mathematics, and linguistics to name a few. In computer science, for example, grammars can be used to represent certains kinds of limited computers, known as deterministic finite automata or push-down automata. In linguistics, formal grammars can be used to parse sentences into parts of speech, such as noun phrases, adjectives, etc. But they also have applications in art as L-systems are excellent tools for generating fractals.

For this problem, consider the following L-system:

variables: A B

constants: none

start: A

rules: (A → AB), $(B \rightarrow$ A)

To produce strings from this grammar, we start with the start variable A and apply the rule continuosly.

```
n=0:         A           start (axiom/initiator)
            / \
n=1:       A   B         the initial single A spawned into AB by rule
          /|    \           (A -> AB), rule (B -> A) couldn’t be applied
n=2:     A B     A       former string AB with all rules applied, A spawned into
        /| |     |\         AB again, former B turned into A
n=3:   A B A     A B     note all A’s producing a copy of themselves in the
      /| | |\    |\ \       first place, then a B, which turns ...
n=4: A B A A B   A B A   ... into an A one generation later, starting to spawn/repeat/recurse then
```

The above was taken from `wikipedia.org`.

As a string, the production looks like

```
n=0:  A
n=1:  AB
n=2:  ABA
n=3:  ABAAB
n=4:  ABAABABA
```

Show that the total number of nodes in the tree produced by the given L-system is $\Omega(2^{\frac{n}{2}})$.

2-3 Trees

Assume that you are given a sorted array of n elements. Explain how to create in O(n) time a 2-3 tree where every non-leaf node has exactly 3 children. Assume $n = 2(1 + 3 + 3^2 + 3^3 + \cdots + 3^h)$ for some integer value h.

Note that to proving this would take lots of writing. So, a full proof is not required. You are allowed to give a more hand-wavy explanation. You may use pictures to help illustrate your idea. Basically, the answer will be considered correct if you can give an explanation that gives the main idea of how to solve the problem.