Math/CCS 103 Professor: Padraic Bartlett

Homework 4: Presentations from Week 2

Due Friday, week 3 *UCSB 2014*

Homework Problems.

Pick **three** of the problems below, and solve them!

1. (Alex) Evaluate ${}^{3}3$. Estimate the number of digits in ${}^{4}3$ and ${}^{3}4$. Do you expect ${}^{10}11$ or ${}^{11}10$ to be greater? Justify your answer.

2. (Alex) There are some values of x for which ${}^{\infty}x$ is a finite number. 1 is a simple example, because 1 raised to any power is still 1. Either show that for no values $x > 1$ can ${}^{\infty}x$ have a finite value; or if not, find an example of an x that *does* remain finite, show that it does, and evalulate ${}^{\infty}x$ numerically.

3. (Alex) Besides some expressions like $\frac{0}{0}$ or $\ln(0)$, the power 0^0 is far less recognized as something ill-defined. On the one hand, 0 to any positive power is 0, so $0^0 = 0$. On the other hand, anything to the power 0 must be 1, so $0^0 = 1$. This can be formalized as

$$\lim_{x \to 0} 0^x = 0$$

$$\lim_{x \to 0} x^0 = 1$$

Tetration lets us resolve this problem, however – or at least throw another possible solution into the mix. Noting that $0^0 = {}^{2}0$, find the limit $\lim_{x \to 0} {}^{2}x$, and this should give us an answer!

4. (Alex) Extending the previous problem, examine the behavior of

$$\lim_{x \to 0} {}^{3}x$$

$$\lim_{x \to 0} {}^{4}x$$

and try to come up with a general formula for

$$\lim_{x \to 0} {}^{n}x$$

5. In a graph $G = (V, E)$, a **Hamiltonian cycle** is a sequence of vertices and edges $(v_1, e_{12}, v_2, e_{23}, \ldots v_n, e_{n1}$, such that

   - each vertex in V shows up in our sequence exactly once, and
   - the edges e_{ij} are all edges linking vertex v_i to vertex v_j.

   In other words, a Hamiltonian cycle is a tour that starts and stops at the same vertex, and along the way visits every other vertex exactly once.

(a) Find an algorithm that takes in a graph on n vertices and outputs "Y" if it has a Hamiltonian cycle, and "N" if it does not.

(b) Find a reasonable upper bound on the runtime of your algorithm. (It should be really big.)

(c) Show that your problem is in NP: i.e. find an describe that will take in an instance of this "pathing" problem and any "proof" that claims to show that instance is true, and check in polynomial time whether that solution holds.

6. Take a graph G. We can play a solitaire game, called **pebbling**, on this graph. We define this as follows:

- Setup: a graph G. Also, to every vertex of G, we assign some number of "pebbles," which we imagine are stacked on top of each vertex.
- Moves: Suppose we have an edge e_{12} connecting v_1 to v_2, and another edge e_{23} connecting v_2 and v_3. Suppose further that there is a pebble on v_1 and v_2. We can then "jump" the pebble v_1 over the pebble at v_2 to v_3: i.e. we can remove one pebble from each of v_1 and v_2, and place a pebble on v_3.
- A graph is **cleared** if it has at most one pebble on its board; similarly, we say that a graph is **clearable** if there is some sequence of moves that clears it.

(a) Find an algorithm that takes in a graph on n vertices with some number m of pebbles on its vertices, and outputs "Y" if it is clearable, and "N" if it is not.

(b) Find a reasonable upper bound on the runtime of your algorithm. (It should still be really big.)

(c) Show that your problem is in NP: i.e. describe an algorithm that will take in an instance of this "clearing" problem and any "proof" that claims to show that instance is true, and check in polynomial time whether that solution holds.