CSc 130 Homework Part 1

Dr. Jinsong Ouyang

September 2017

1 Math Reivew

1.1 Prove the following by Induction

- $\sum_{i=0}^{N} 2^i = 2^{N+1} - 1$
- $\sum_{i=0}^{N} A^i = \frac{A^{N+1}-1}{A-1}$
- $\sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6}$
- $\sum_{i=0}^{\infty} A^i = \frac{1}{1-A}$, if $0 < A < 1$
- $F_n > (\frac{3}{2})^n$, $\forall F_n$, a fibonacci number, where $n \geq 5$

1.2 Compute the the following

- $\sum_{i=0}^{\infty} \frac{1}{4^i}$
- $\sum_{i=0}^{\infty} \frac{i}{4^i}$

2 Relative Complexity

Relative complexity can be computed using,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \begin{cases} 0, & \implies f(n) = O(g(n)) \\ c, & \implies f(n) = \Theta(g(n)) \\ \infty, & \implies f(n) = \Omega(g(n)) \\ DNE, & \implies \text{There is no relationship between } f(n) \text{ and } g(n). \end{cases}$$

Use the above technique, and sort the following functions' complexity in an increasing order.

$$2^n, 2^{1000000}, n \log n, \frac{n^{\frac{1}{1000}}}{(\log n)^{1000}}, n^{\frac{1}{2000}} (\log n)^{1000}, (\log n)^n, (\log n)^{1000000}, \frac{n^{\frac{1001}{1000}}}{(\log n)^{1000}}$$

3 Trees

3.1 Algorithm design

1. Design an algorithm (pseudo code) to find the height of a binary tree.
2. Design an algorithm (pseudo code) to return the longest path from the root to a leaf node in a binary tree.
3. Design an algorithm (pseudo code) to swap the children for each node in a binary tree.

3.2 Proof

1. For a binary tree of height h, there is a maximum of 2^h leaf nodes in the tree.
2. For a binary tree of height h, there is a maximum of $2^{h+1} - 1$ nodes in the tree.