COS226 Week 1 Activity

1. *Empirical analysis. Algorithms textbook 1.4* The following table gives approximate running times for a program with N inputs, for various values of N.

N	time
500	2.00 seconds
1000	4.44 seconds
2000	9.77 seconds
5000	27.37 seconds

   Predict its running time (in minutes) for $N = 10,000$ and give a formula that estimates the running time as a function of N.

2. Suppose you're trying to estimate running time. What is good about the data above, but wrong with the data below? Give at least 2 reasons why the data above is superior.

N	time
10	0.0030 seconds
40	0.013 seconds

3. Give a formula that estimates the running time (in seconds) in terms of N and M for the program whose timing data is given below.

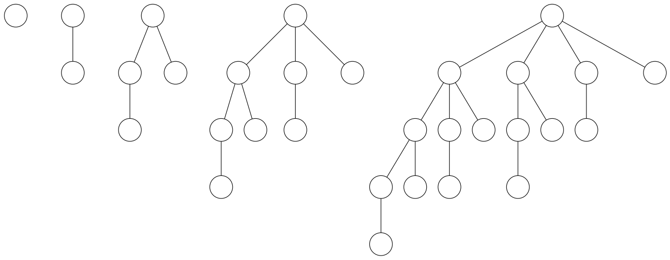
Table for M = 1000

N	times
100	2.00 seconds
200	4.03 seconds
400	8.05 seconds
800	16.21 seconds

Table for N = 100

M	times
1000	2.01 seconds
2000	8.02 seconds
4000	32.54 seconds
8000	128.65 seconds

4. *Worst-case input for weighted quick-union. Algorithms textbook 1.5*
A *binomial tree* is defined recursively: a binomial tree of order 0 consists of a single node; a binomial tree of order h is a tree obtained from two binomial trees of order $h-1$, by linking the root of one to the other. Below are binomial trees of order 0, 1, 2, 3, and 4. The height of a tree is the maximum number of links that must be traversed to reach the root from the bottom.



(a) How many nodes are in a binomial tree of order h?

(b) And what is the height of a binomial tree of order h?

(c) What is the minimum number of `union()` operations (using the weighted quick-union algorithm) that produces a binomial tree of order $h = 3$.

(d) What is the worst case number of array accesses of `find()` on a binomial tree, as a function of its number of nodes N?

```
public int find(int p) {
    while (p != id[p])
        p = id[p];
    return p;
}
```