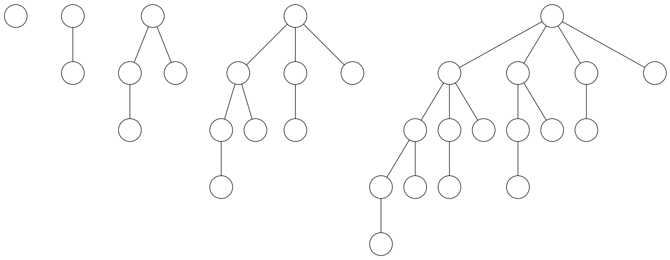
COS226 Week 1 Activity

1. The code below is part of the weighted quick-union algorithm. Give a crisp and concise English description of the `union()` algorithm—don't write Java code.

```
public void union(int p, int q) {
    int i = find(p);
    int j = find(q);
    if (i == j) return;

    // make smaller root point to larger one
    if   (sz[i] < sz[j]) { id[i] = j; sz[j] += sz[i]; }
    else                 { id[j] = i; sz[i] += sz[j]; }
}
```

2. *Worst-case input for weighted quick-union.* 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.



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

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

(c) Give a sequence 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;
}
```

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

N	time
1000	10 seconds
2000	40 seconds
5000	~4 minutes

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