

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	2.5 seconds
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 .

2. What is good about the data above? What's wrong with the data below? Give at least 2 reasons why.

N	time (seconds)
10	0.0030
40	0.013

3. *Memory analysis. Algorithms textbook 1.4* Suppose you have an array `p[]` as declared and initialized below.

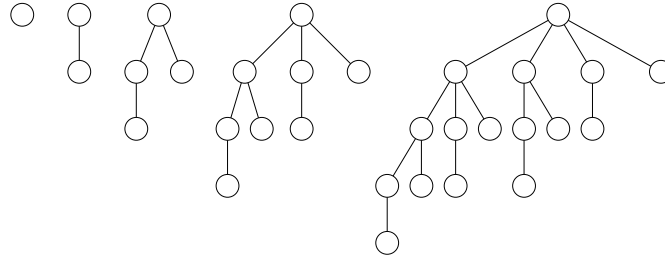
- How much memory (in bytes) does the array use as a function of N ? Include the memory for both the array and the points.

```
public class Point {
    private final int x;
    private final int y;
}
Point[] p = new Point[N];
for (int i = 0; i < N; i++)
    p[i] = new Point(...);
```

- Repeat the previous question, but use tilde notation to simplify your answer.

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.



- (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 order of growth 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;  
}
```