Order Statistics Review

Lecture 8

CS321

Todays Lecture

- Assignment 2 is due Tuesday Feb 17th 11:59pm
- HW1 out on Monday Feb 15th.
- Monday 15th is President's Day: no class.

Sorting Compared

Method/Structure	Space	Complexity	Time
Insertion Sort/Array	O(n)	Simple	$O(n^2)$
Quicksort/Array	O(2*n)	Complex	$O(n^2)$
Heapsort/Tree	O(n)	Complex	O(n*log(n))
Mergesort/Array	O(2*n)	Simple	O(n*log(n))
Counting Sort/Array	O(2*n)	Simple	O(3*n)

Big O Notation

- f(n) = O(n) means f(n) <= c*n for some c > 0
- f(n) = Ω(n) means f(n) >= c*n for some c > 0.
- f(n) = Θ(n) means f(n) = O(n) and f(n) = Ω(n)

- Sources:
  - Introduction to Algorithms Chapter 2
  - http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap02.htm
  - Not covered in OpenDataStructures: need CLRS

Challenge Question

- For breakout groups:
  - Is $2^{n+1} = O(2^n)$
  - Is $2^{2n} = O(2^n)$

Calculate the O for this function

- f(n) = Θ(n)

```
i = 1;
sum = 0;
while (i <= n)
        do if (f(i) > k)
                then sum += f(i);
             i = 2*i;
```

What does this mean?

The running time of f(n) is at least $O(n^3)$.

$$\Theta(n) + \Omega(n) = \Theta(n)$$

$$f(n) = O(g(n)) \text{ implies } g(n) = O(f(n))$$

Order the following

$$\log_2(n!),\ 4^n,\ \log_2 n,\ \log_2 n^n,\ \log_{10} n,\ \log_{10} n^{10},\ \sqrt{n},\ 5^{n/2}, n!$$

Bolt Sort – AKA Quicksort

- Problem: Sorting a set of Nuts and Bolts
- You have N nuts, and N bolts.
- For every bolt there is a matching nut.
- You can test a bolt against a nut.
- You can not compare a bolt to a bolt.
- You can not compare a nut to a nut.
- What is an efficient way to match all bolts and nuts?