Note Title 10/15/2007

Randomized Algorithms

What does it mean for a value to be randomly selected?

How can we make use of randomness?

Monte Carlo Algorithms (451)

- Don't always give the correct answer.
- The runtime can be described consistently.

Las Vegas Algorithms (351)

- They always give the correct answers.
- Their runtime is not consistent.

Random Median Finding #1

Algorithm

- Select a value at random, call it *p*.
- Partition the list around *p*.
- See if it was the median (same number in each side of the partitioning).
- If it is, great. If it wasn't, oh well, try again…

$$\left.\begin{array}{l} n-1 \\ n-2 \\ n-3 \\ n-4 \end{array}\right\} n$$

Question #1: Does this work?

Question #2: Is it a good algorithm?

Random Median Finding #2

Algorithm

- Select a value at random, call it *p*.
- Partition around *p*.
- See if it was the median (same number in each side of the partitioning).
- If it wasn't, then we have still found the x^{th} smallest value in the list (the value of x will be based on the size of the partitions).
  - If x is "before" the median, take the right side and find the $(n/2-x)^{th}$ smallest.
  - Otherwise, take the "left" side and find the $(n/2)^{th}$ smallest.

Note: If this ends up being a good idea, we'd end up coding general selection.

Question #1: Does this work?
Question #2: Is it a good algorithm?

Compute Runtime

How do we analyze the runtime of something like this?

Partitioning takes n-1 comparisons.
Random number generation may take some time (we'll ignore this for now).
The recursion may or may not be needed, and we don’t know exactly how many values will be passed into that recursion.

$$T(n) = n + T(???)$$

The best case is easy, we find it on the first shot and it’s n-1 comparisons. $O(n)$
What about worst case and average case?

$$T(n) = n + T(n-1)$$

n
$n-1$
$n-2$
$n-3$
$\vdots$
$O(n^2)$

"Weighted" average of all possible splits

Expected Runtime

We will assume unique values in the list.

We'll round things and say the partitioning takes n comparisons.

We will look at "worst" expected runtime.

We'll compute assuming we have to look in the larger of the two sub-lists (which is true for median finding).

We won't worry about floor/ceiling issues in this initial exploration.

Let pos_p be the position of the partition value.

We know if pos_p is between 1 and $\frac{n}{2}$, then $n - pos_p$ is bigger.

And if pos_p is between $\frac{n}{2}$ and n, $pos_p - 1$ is bigger.

$$T(n) \le n + \frac{\sum_{pos_p=1}^{n} T(\max(pos_p - 1, n - pos_p))}{n}$$

Assume equal likelihood $\longrightarrow n$

How can we simplify this?

We know the two sizes are symmetric.

So we can rewrite the summation above:

$$T(n) \le n + \frac{2 \sum_{pos_p=1}^{\frac{n}{2}} T(n - pos_p)}{n} \quad \left\{ \begin{array}{c} T(n-1) \\ T(n-2) \\ T(n-3) \\ \vdots \\ T(\underbrace{n - \tfrac{n}{2}}_{\frac{n}{2}}) \end{array} \right\}$$

What is $n - pos_p$?

Let $size = n - pos_p$

$$T(n) \le n + \frac{2}{n} \left(\sum_{size=\frac{n}{2}}^{n-1} T(size) \right)$$

How big is this?

Let's guess $T(n) \le cn$.

Use induction to see if we are right.

I gave in and used "n" instead of "k". ☺

Assume $\forall i < n,\ T(i) \le ci$

Show $T(n) \le cn$

$$T(n) \le n + \frac{2}{n}\sum_{size=\frac{n}{2}}^{n-1} T(size) \overset{?}{\le} cn$$

$$n + \frac{2}{n}\sum_{size=\frac{n}{2}}^{n-1} \underbrace{c \cdot size}_{\text{I.H.}} \overset{?}{\le} cn$$

Note: size is a value that is less than n, so we can apply I.H.

$$n + \frac{2c}{n}\underbrace{\sum_{size=\frac{n}{2}}^{n-1} size}_{\text{upper piece}} \overset{?}{\le} cn$$

$$n + \frac{2c}{n}\left(\sum_{s=1}^{n-1} s - \underbrace{\sum_{s=1}^{n/2-1} s}_{\text{Subtract off lower piece}}\right) \overset{?}{\le} cn$$

$$n + \frac{2c}{n}\left(\frac{(n-1)(n)}{2} - \frac{\left(\frac{n}{2}-1\right)\left(\frac{n}{2}\right)}{2}\right) \overset{?}{\le} cn$$

$$n + \frac{c}{n}\left(n^2 - n - \left(\frac{n}{2}-1\right)\left(\frac{n}{2}\right)\right) \overset{?}{\le} cn$$

$$n + \frac{c}{n}\left(n^2 - n - \left[\frac{n^2}{4} - \frac{n}{2}\right]\right) \overset{?}{\le} cn$$

$$n + c\left(n - 1 - \frac{1}{4}n + \frac{1}{2}\right) \overset{?}{\le} cn$$

$$n + cn - c - \frac{1}{4}cn + \frac{1}{2}c \overset{?}{\leq} cn$$

$$n + \frac{3}{4}cn - \frac{1}{2}c \overset{?}{\leq} cn$$

$$n + \frac{3}{4}cn - \frac{1}{2}c < \underbrace{n + \frac{3}{4}cn \overset{?}{\leq} cn}_{\text{New Goal}}$$

$$n \overset{?}{\leq} \frac{1}{4}cn$$

$$1 \overset{?}{\leq} \frac{1}{4}c$$

$$c \geq 4$$

Expected Value is at worst $4n$ so $O(4n)$ expected time!