CS240 - Tutorial 4

Graham Cooper

May 27th, 2015

QuickSelect Example

Q1

Find the 3rd smallest element in A
Pivot = A[0]
A =

8	17	10	1	6	20	2	9	7	13
<u>8</u>	<u>17</u>	10	1	6	20	2	9	<u>7</u>	13
8	7	10	1	6	20	2	9	17	13
8	17	<u>10</u>	1	6	20	<u>2</u>	9	7	13
8	7	2	1	6	20	10	9	17	13

Place Pivot

6	7	2	1	8	20	10	9	17	13

Recurse on the left hand side (6,7,2,1)
pivot = A[0]

6	<u>7</u>	2	<u>1</u>
6	1	2	7

2	1	6	7

Recurse on left hand side (2, 1)

Since the pivot is at index 3, it correspons to the 3rd smallest element, we are done.

Q2)

Assume A has distinct elements.

```
Bogo(A){
shuffle(A) //O(n)
if A is sorted{
return A;
}
else {
```

```
return Bogo(A);
}
}
```

Best case is O(n)
Worst case is O(∞)
May not terminate

$$T_{avg}(n) = 1 \cdot cn + \underbrace{(\frac{1}{n!} \cdot d)}_{Aissorted} + \underbrace{((1 - \frac{1}{n!}))T_avg(n)}_{Aisnotsorted}$$

$$T_{avg}(n)[1 - (1 - \frac{1}{n!})] = cn + \frac{1}{n!} \cdot d$$

$$T_{avg}(n) = cn \cdot n! + d \in O(n \cdot n!)$$

EE[x] = $\sum_{x \in X} P_r(x) \cdot RunningTime(X)$

Q3)

Toss identical balls at random into buckets (or bins), one at a time, uniformly at random. How many tosses can we expect to make such that every bucket contains at least 1 ball.

- Define a toss in which a ball falls into an empty bucket as a hit and a non-empty bucket as a miss
- Partition the tosses into stages
– The i^{th} stage consists of the tosses after the $(i-1)^{th}$ hit until (And including) the i^{th} hit.

Ex. 4 Buckets
Toss sequence: 2|,2,3|,4|,3,3,2,4,1|

During the i^{th} stage.
- (i-1) non-empty buckets
- (b - i + 1) empty buckets (b is the bumber of buckets)
- Pr(throwing in empty bucket) = $\frac{b-i+1}{b}$

Define n_i = number of throws in stage i

$$n = \sum_{i=1}^{b} n_i$$

$$EE[n] = EE[\sum_{i=1}^{b} n_i]$$

$$= \sum_{i=1} bEE[n_i]$$

the above by linearity of expectation
$n_i \approx$ geometric dist.
EE[n_i] = 1/p

$$= b \sum_{i=1}^{b} \frac{1}{b-i+1}$$

$$= b \sum_{i=1}^{b} \frac{1}{i} \in \Theta(blnb)$$

Q4

Argue that any comparison based sorting algorithm requires at least 7 comparisons to S numbers

- An algorithm performs actions as the result of different comparisons.

If (A[i] < A[j]) then do stuff, else something else

- elements, assume worst case that we have a permutation for each leaf in a tree, therefore 5! leaves
- height of any binary tree on 5! leaves will be log(5!) $\approx$ 6.8 < 7

Expected Time Analysis

Probability Review

Intro to Lower Bounds