Institut for Matematik og Datalogi
Syddansk Universitet

September 28, 2020
JFB

DM551/MM851 – Algorithms and Probability – 2020
Lecture 6

Lecture, September 23

We finished section 7.2 and covered Baye's Theorem in section 7.3 (you should read the rest of that section to discuss in discussion section). We began on section 7.4, covering the defintion of the expected value of a random variable.

Lecture, September 28

We will almost finish section 7.4 (Example 9 will be done using the InsertionSort algorithm on this lecture note, along with the analysis there. This is the preferred InsertionSort for this course.)

Lecture, October 1

We will finish section 7.4, with an example using Chebyshev's Inequality. We will cover sections 8.5 and 8.6.

Problems to be discussed on October 5/6

1. Finish any problems not finished on September 30.
2. Section 7.4: 28, 29, 37 (prepare at home and bring to class), 38, 43 (Additional hint: Show that $E[X_i \cdot X_j] = \frac{1}{n(n-1)}$).
3. Supplementary Exercises, Chapter 7 (pages 522–523): 10, 13, 21.

Average case complexity of Insertion Sort

procedure InsertionSort(List):
{ Input: List is a list }
{ Output: List, with same entries, but in nondecreasing order }

> N := 2
> **while** ($N \leq$ length(List)
> **begin**
>> Pivot := Nth entry
>> j := $N - 1$
>> **while** ($j > 0$ and jth entry $>$ Pivot)
>> **begin**
>>> move jth entry to loc. $j + 1$
>>> j := $j - 1$
>>
>> **end**
>> place Pivot in $j + 1$st loc.
>> N := $N + 1$
>
> **end**

We count the number of comparisons. Let n be the number of elements in the list (length(List)).

For the worst case, consider a list originally in reverse order. For each value of N from 2 to n, the Nth entry is compared to all $N - 1$ entries before it in the list. This gives $\sum_{N=2}^{n}(N-1) = \sum_{j=1}^{n-1}(j) = \frac{n(n-1)}{2}$. Since the Nth entry is never compared to more than all of the $N - 1$ entries before it in the list, the worst case is when the list was in reverse order, and the worst case number of comparisons is exactly $\frac{n(n-1)}{2} = \frac{n^2-n}{2}$.

For the average case, we assume a random ordering (permutation) of the entries in the list originally.

Let the random variable X be the number of comparisons done by the algorithm. Let the random variable X_i be the number of comparisons to insert entry N after the first $N - 1$ entries are already sorted.

$$X = X_2 + X_3 + \cdots + X_n$$

By the linearity of expectations,

$$E[X] = E[X_1] + E[X_2] + \cdots + E[X_n]$$

For $0 \leq k \leq N - 1$, let $p_N(k)$ be the probability that entry N gets placed in location $N - k$ (just after the first $N - 1$ entries are sorted). All locations are equally likely, so $p_N(k) = \frac{1}{N}$. If entry N gets placed in location $N - k$, it was compared with $k + 1$ entries, before finding one not larger than itself. Now we can compute the expectation of each X_N.

$$\begin{array}{rcl} E[X_N] & = & \sum_{k=0}^{N-1} p_N(k) \cdot (k+1) \\ & = & \sum_{k=0}^{N-1} \frac{1}{N} \cdot (k+1) \\ & = & \frac{1}{N} \sum_{j=1}^{N} j \\ & = & \frac{N+1}{2} \end{array}$$

Thus,

$$\begin{array}{rcl} E[X] & = & \sum_{N=2}^{n} E[X_N] \\ & = & \sum_{N=2}^{n} \frac{N+1}{2} \\ & = & \frac{1}{2} \sum_{j=3}^{n+1} j \\ & = & \frac{1}{2} \left(\frac{(n+1)(n+2)}{2} - (1+2) \right) \\ & = & \frac{n^2+3n-4}{4} \end{array}$$

So this is a factor less than 2 better than worst case. Note, however, how well this algorithm does on sorted (or nearly sorted) lists.