Institut for Matematik og Datalogi
Syddansk Universitet

September 13, 2018
JFB

DM551 – Algorithms and Probability – 2018
Lecture 3

Lecture, September 11

We covered sections 7.2 and 7.3.

Lecture, September 17

We will cover section 7.4 (Example 9 will be done using the Insertion Sort algorithm on this lecture note, along with the analysis there. This is the preferred InsertionSort.)

Lecture, September 24

We will cover sections 8.5 and 8.6.

Problems to be discussed on September 25

1. Section 7.4: 6, 10, 22, 23, 27 (important), 28, 33 (Additional hint: Show that $E[X_i \cdot X_j] = \frac{1}{n(n-1)}$).
2. Supplementary Exercises, Chapter 7 (pages 481–482): 10, 12, 18.
3. Do problem 2 in this exam set written by Jørgen Bang-Jensen in Danish:

   http://imada.sdu.dk/∼jbj/DM551/jan09.pdf

   Here it is in English:

   A particle starts at time $t = 0$ at a point O (consider it point $x = 0$ on the x-axis). At every time unit, the particle either moves one unit forward (in the positive x direction) with probability p, or stays still with probability $q = 1 - p$. Let $P_n(r)$ be the probability that the particle is in position r (on the x-axis) at time $t = n \geq r \geq 0$.

   - Show that $P_n(r) = \binom{n}{r} p^r q^{n-r}$.

- Suppose now that the particle is in the (x, y)-coordinate system, and it starts at $O = (0, 0)$. Now suppose that at each time unit, it moves one unit in the positive x-direction with probability p, or moves one unit in the positive y-direction with probabilty $q = 1-p$. What is the probability $Q_n(r, s)$ that the particle is at the coordinates (r, s) at time $t = n \geq r, s \geq 0$?
- Suppose that $p = 1/3$ and $q = 2/3$ in the above scenario (2-dimensional). What are the probabilities of the following events?
  - The particle passes through $(5, 2)$ (at some time).
  - The particle passes through $(5, 2)$ and $(7, 1)$.
  - The particle passes through $(5, 2)$ and $(6, 3)$.

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 ≤ 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 + 1st 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 $0 \le k \le N-1$, 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.

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

Thus,

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

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.