A trivial code whose analysis isn’t – An exercise in average-case analysis

K. Viswanathan Iyer
Dept. of Computer Science and Engineering
National Institute of Technology
Tiruchirapalli – 620 015
Tamil Nadu
Finding the maximum in a list
Average-Case Analysis

Tools/Technique(s) we require:

2. Basic Probability Theory.
$X[1..n]$ – An array of $n$ distinct positive real numbers.

The following pseudo-code segment FindMax returns the maximum element in $X$.

```
max := -1;
for i := 1 to n do
  if max < X[i] then max := X[i];
```

Problem: What is the Average-case time complexity of FindMax?

This amounts to finding the average number of times the assignment “max := X[i]” is executed.
Rewriting FindMax

FindMax can be written as the following assembly-language-like code using a reduced set of pseudo-code statements:

```
max := -1 ;      
i := 0 ;        
1:   i := i + 1 ;
    if i > n then goto 2 ;
    if max >= X[i] then goto 1 ;
    max := X[i] ;
    goto 1 ;
2:   . . .
```
Executing \texttt{FindMax}

On a sequential computer, \texttt{FindMax} will execute:

- a fixed number of assignments to initialize $\texttt{max}$ and $\texttt{i}$.
- $(n + 1)$ comparisons of the form "$\texttt{i} > n$?".
- $(n + 1)$ increments of the index $\texttt{i}$.
- $n$ comparisons of the form "$\texttt{max} \geq \texttt{X[i]}$".
- a variable number (between $n_1$ and $n_2$) of assignments "$\texttt{max} := \texttt{X[i]}$".

Thus the time $t_{\texttt{maxf}_n}$ taken by \texttt{FindMax} is of the form:

$$t_{\texttt{maxf}_n} = c_0 + c_1 n + c_2 \texttt{EXCH}[\texttt{X}],$$

where $\texttt{EXCH}[\texttt{X}] = \text{number of times the instruction } \texttt{"max} := \texttt{X[i]}" \text{ is executed}$;
$c_0$, $c_1$, $c_2$ are implementation constants dependent on the machine where the code executes.
What are $n_1$ and $n_2$?
The Permutation Model

To estimate the expected value of $EXCH[X]$, we introduce the “permutation model”:

We assume that the array $X$ is a permutation of the integers $(1, \ldots, n)$. We also assume that each permutation is equally likely (to be an input to $\text{FindMax}$). Thus each permutation can occur with a probability $\frac{1}{n!}$. Let $s_{n,k}$ = number of those permutations wherein $EXCH[X] = k$. Let $p_{n,k}$ = probability that $EXCH[X] = k$. Then

$$p_{n,k} = \frac{s_{n,k}}{n!}.$$  

Then the expected value $exch[X]$ of $EXCH[X]$ is given by

$$exch[X] = \sum_{k=1}^{n} k \frac{s_{n,k}}{n!} = \frac{1}{n!} \sum_{k=1}^{n} ks_{n,k}. \quad (1)$$
Getting the sum on the right-hand-side of (1) above

We consider all those permutations $\sigma_1, \sigma_2, \ldots, \sigma_n$ of $(1, 2, \ldots, n)$, wherein $EXCH[k] = k$ – by definition, there are $s_{n,k}$ of these.

With respect to these permutations, we have the following two mutually exclusive (and totally exhaustive) cases:

1. $\sigma_n = n$ : in this case $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ should have produced exactly $k-1$ exchanges – the number of permutations in this case is $s_{n-1,k-1}$.

2. $\sigma_n \neq n$ : in this case the last element is one of $1, \ldots, (n-1)$; then $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ should have produced exactly $k$ exchanges – the number of permutations in this case is $(n-1)s_{n-1,k}$.

Thus we have

$$s_{n,k} = s_{n-1,k-1} + (n-1)s_{n-1,k}. \quad (2)$$
How to get to the goal?

Recall (2): \( s_{n,k} = s_{n-1,k-1} + (n - 1)s_{n-1,k} \).

We introduce the following generating function \( S_n(x) \) for each \( n \):

\[
S_n(x) = \sum_{k=1}^{n} s_{n,k} x^k. \tag{3}
\]

Boundary case: it follows that \( S_1(x) = x \).
We multiply both sides of (2) by \( x^k \) and sum over \( k \) from 1 through \( n \). We then invoke definition (3) to get

\[
S_n(x) = xS_{n-1}(x) + (n - 1)S_{n-1}(x) = (x + n - 1)S_{n-1}(x). \tag{4}
\]

Using the above boundary condition in (4), we get \( S_2(x) = x(x + 1) \). In general, it follows that the explicit form for \( S_n(x) \) is given by

\[
S_n(x) = \prod_{j=0}^{n-1}(x + j). \tag{5}
\]
More manipulations ... 

From the definition (3) of $S_n(x)$ we get

$$S'_n(x) = \sum_{k=1}^{n} ks_{n,k}x^{k-1},$$

which gives

$$S'_n(1) = \sum_{k=1}^{n} ks_{n,k}. \quad (6)$$

Also, from the explicit form of $S_n(x)$ in (5) we get

$$S_n(1) = \prod_{j=0}^{n-1}(1 + j) = n!. \quad (7)$$

Using (6) and (7) in the expression for $exch[X]$ in (2), we get

$$exch[X] = \frac{1}{n!} \sum_{k=1}^{n} ks_{n,k} = \frac{S'_n(1)}{S_n(1)}. \quad (8)$$
The derivative (w.r.t. $x$) of (5) after taking logarithm on both sides gives

$$\frac{S'_n(x)}{S_n(x)} = \frac{1}{x} + \frac{1}{x+1} + \ldots + \frac{1}{x+n-1}. \quad (9)$$

We substitute $x = 1$ in (9). This gives

$$\frac{S'_n(1)}{S_n(1)} = H_n, \quad (10)$$

where $H_n$ is the $n^{th}$ Harmonic number, which is the value of $\text{exch}[X]$.

We have thus proved the following:

**Theorem:**
Under the permutation model, the average-time $t_{maxf}^{avg}$ taken by $\text{FindMax}$ is given by

$$t_{maxf}^{avg} = c_0 + c_1n + c_2H_n. \quad (11)$$