Mark Watson
CS4720
Homework #1
09/12/2011

**Problem 1** Sum the following series to obtain closed form solutions.

- ($a$) The series: $1^2 + (1^2 + 2^2) + (1^2 + 2^2 + 3^2) + \dots$ to n terms.
- (b) The series: $4 + 44 + 444 + \dots$ to n terms.
- (c) The series: $1 + 4x + 12x^2 + 32x^3 + \dots$ when $0 < x < \frac{1}{2}$

(a) Summations for $\sum_{k=1}^{n} k^2$ and$\sum_{k=1}^{n} k$ where taken from page 470 of Levitin's book.

$$\begin{aligned}
& 1^2 + (1^2 + 2^2) + (1^2 + 2^2 + 3^2) + \dots \\
= & \sum_{k=1}^{1} k^2 + \sum_{k=1}^{2} k^2 + \sum_{k=1}^{3} k^2 + \sum_{k=1}^{4} k^2 + \dots \\
= & \sum_{k=1}^{n} \sum_{j=1}^{k} j^2 \\
= & \sum_{k=1}^{n} [\tfrac{1}{6}(k(k+1)(2k+1))] \\
= & \tfrac{1}{6} \sum_{k=1}^{n} [(k^2+k)(2k+1))] \\
= & \tfrac{1}{6} \sum_{k=1}^{n} [2k^3 + 3k^2 + k] \\
= & \tfrac{1}{6} [2 \sum_{k=1}^{n} k^3 + 3 \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} k] \\
= & \tfrac{1}{6} [2 (\sum_{k=1}^{n} k)^2 + 3 \sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} k] \\
= & \tfrac{1}{6} [2(\tfrac{n^2(n+1)^2}{4}) + \tfrac{3}{6}(n(n+1)(2n+1)) + \tfrac{1}{2}n(n+1)] \\
= & \tfrac{1}{6} \left( \tfrac{1}{2}n(1+n) + \tfrac{1}{2}n^2(1+n)^2 + \tfrac{1}{2}n(1+n)(1+2n) \right) \\
= & \tfrac{1}{12} n(1+n)^2(2+n)
\end{aligned}$$

(b) Summation formula 5 was used from Levitin's book pg. 470 to solve summations in the form $\sum_{k=0}^{n} 10^k$.

$$
\begin{aligned}
& 4 + 44 + 444 + \dots \\
= \quad & 4(1 + 11 + 111 + \dots) \\
= \quad & 4 \sum_{k=1}^{n} \sum_{j=1}^{k} 10^{(j-1)} \\
= \quad & 4 \sum_{k=1}^{n} \sum_{j=0}^{k-1} 10^{j} \\
= \quad & 4 \sum_{k=1}^{n} \frac{10^{(k-1)+1} - 1}{10 - 1} \\
= \quad & \frac{4}{9} \sum_{k=1}^{n} 10^k - 1 \\
= \quad & \frac{4}{9} \left( \sum_{k=1}^{n} 10^k - \sum_{k=1}^{n} 1 \right) \\
= \quad & \frac{4}{9} \left( \sum_{k=1}^{n} 10^k - n \right) \\
= \quad & \frac{4}{9} \left( \left( \sum_{k=0}^{n} 10^k \right) - 1 - n \right) \\
= \quad & \frac{4}{9} \left( \left( \frac{10^{n+1} - 1}{10 - 1} \right) - 1 - n \right) \\
= \quad & \frac{4}{9} \left( \frac{10^{n+1} - 1}{9} - 1 - n \right) \\
= \quad & \frac{4}{9} \left( \frac{10^{n+1} - 1 - 9 - 9n}{9} \right) \\
= \quad & \frac{4}{81} (10^{n+1} - 10 - 9n)
\end{aligned}
$$

(c) The series can be expressed as summation.

$$\begin{aligned} & 1 + 4x + 12x^2 + 32x^3 + \cdots + n2^{n-1}x^{n-1} \quad \textit{when}\, 0 < x < \frac{1}{2} \\ = & \quad \textstyle\sum_{k=1}^{n} k2^{k-1}x^{k-1} \\ = & \quad \textstyle\sum_{k=1}^{n} k(2x)^{k-1} \end{aligned}$$

Subtracting $(2x)$ times series from the original series to find the sum:

$$\begin{aligned} S &= 1 + 4x + 12x^2 + 32x^3 + 80x^4 + \ldots \\ -(2x)S &= \quad\;\; 2x + \;8x^2 + 24x^3 + 64x^4 + \ldots \\ S(1-2x) &= 1 + 2x + \;4x^2 + \;8x^3 + 16x^4 + \ldots \end{aligned}$$

Which we can represent as the following sum (we make use of equation 5 from Levitin's book on page 470 to solve this summation):

$$\begin{aligned} S(1-2x) &= \sum_{k=0}^{n} 2^k x^k \\ S(1-2x) &= \sum_{k=0}^{n} (2x)^k \\ S(1-2x) &= \frac{(2x)^{n+1} - 1}{(2x) - 1} \\ S &= \frac{(2x)^{n+1} - 1}{(1-2x)(2x-1)} \\ S &= \frac{(2x)^{n+1} - 1}{-(2x-1)^2} \end{aligned}$$

$\blacksquare$

**Problem 2** Show the following by induction:

- (a) Show that $5^{2n} + 3n - 1$is divisible by 9 for any $n \geq 1$.
- (b) Show that $n! > 3^n$ for $n \geq 7, n \in N$.

(a) We can see the hypothesis holds for the *base case* of 1: $5^{2(1)} + 3(1) - 1 = 27$, which is divisible by 9. To show the *induction step*, Let $f(n) = 5^{2n} + 3n - 1$, and assume that $f(k)$ is divisible by 9. Then, we need to show that $f(k+1) - f(k)$ is divisible by 9 to show that $f(k+1)$ is divisible by 9.

$$
\begin{aligned}
& [5^{2(k+1)} + 3(k+1) - 1] - (5^{2k} + 3k - 1) \\
= \quad & 5^{2k}5^2 + 3 - 5^{2k} \\
= \quad & 5^{2k}(5^2 - 1) + 3 \\
= \quad & 25^k(24) + 3
\end{aligned}
$$

Now we need to show that $25^k 24 + 3$ is divisible by 9 to show that $f(k+1)$ is divisible by 9, and that our original hypothesis holds. $\blacksquare$

- *Hypothesis*: $25^n 24 + 3$ is divisible by 9 for all $n \geq 1$
- *Base case*: $25^1 24 + 3 = 603$, which is divisible by 9.
- *Induction step*: Assume that $25^k 24 + 3$ is divisible by 9. Then:

$$
\begin{aligned}
& (25^{k+1}24 + 3) - (25^k 24 + 3) \\
= \quad & 25^{k+1}24 - 25^k(24) \\
= \quad & 25^k(25)(24) - 25^k(24) \\
= \quad & 24(25^k)(25 - 1) \\
= \quad & 25^k(576)
\end{aligned}
$$

Since 576 is divisible by 9, $25^n 24 + 3$ is divisible by 9 for all $n \geq 1$, by induction. Therefore, $5^{2n} + 3n - 1$is divisible by 9 for all $n \geq 1$ by induction.

(b) ∎

- *Hypothesis*: $n! > 3^n$ for $n \geq 7$, $n \in N$
- *Base case*: $7! > 3^7 = 5040 > 2187$.
- *Induction step*: If $k! > 3^k$and $k \geq 7$, then we can show that $(k+1)! > 3^{k+1}$ also holds: $(k+1)! = k!(k+1) > 3^k(k+1)$ (since $k! > 3^k$). Next, $3^k(k+1) > 3^k 3 = 3^{k+1}$, since $k+1 \geq 8 > 3$ ($k > 7$). Therefore, $(k+1)! > 3^{k+1}$, and so our hypothesis holds true by induction.

**Problem 3** Sequential Search Algorithm

The sequential search algorithm is as follows. A is the list being searched, and x is the item we're looking for. ∎

```
def sequential_search(A, x):

    for k=1 to n do

        if A[k] == x then

            return k

        endif

    endfor
    return 0
```

We can analyze the algorithm's best, worst, and average cases using simple summations and assuming each line of code takes "1" time unit to run.

*Best case:* In the best case, x is the first element of the list. The run time in this case would be: 3 time units, or $O(1)$. The best case scenario runs in constant time.

*Worst case:* In the worst case, x is the last element of the list. The run time in this would be: $[\sum_{k=1}^{n} 4] + 1$, assuming that each loop takes four time units, and one additional time unit is needed to return the element. Solving the sum we get: $4n + 1$, or $O(n)$. The worst scenario runs in linear time.

*Average case:* In the average case we assume that the list is ordered in some normal distribution so that an element has an equal probability of being in any part of the list. The run time for this situation would be: $[\sum_{k=1}^{n/2} 4] + 1$, and solving we get: $\frac{4n}{2} + 1 = 2n + 1$, which is still $O(n)$. The average case also runs in linear time. ∎

**Problem 4** Bubble Sort Algorithm.

The pseudo-code for the bubble sort algorithm is as follows, letting A be the list to sort, and n be the length of the list. ■

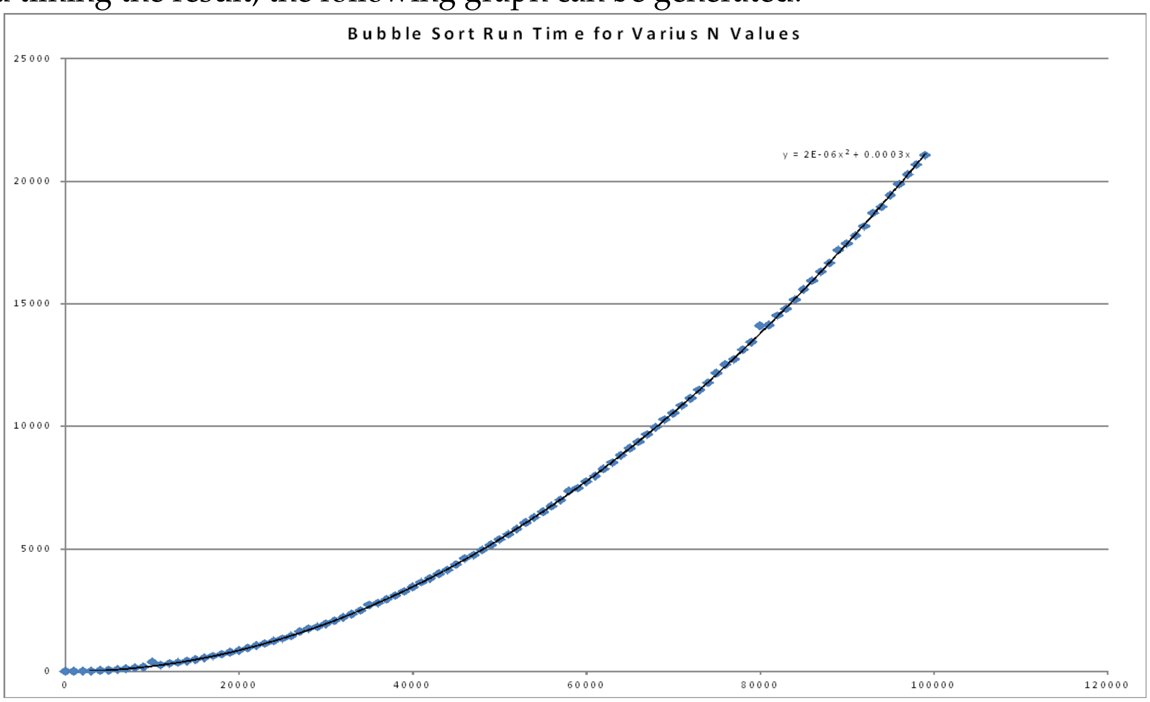
```
for k=1 to n do

    for j=n down to k+1 do
        if A[j] < A[j-1] then
            swap (A[j], A[j-1])
        endif
    endfor

endfor
```

The best, worst, and average cases are all the same, since the code must iterate through the entire list no matter what. There do exist optimizations on the algorithm that break out of the inner loop, however, the algorithm detailed above is the plain vanila one. We can sum up the time to run the algorithm like so (assuming a constant time for the code that's run inside the loop: $\sum_{k=1}^{n-1} k = \frac{1}{2}((n-1)(n-1+1)) = \frac{1}{2}(n-1)n = \frac{1}{2}(n^2-n)$, or $O(n^2)$. So it runs in exponential time.

After running an implementation of the algorithm on lists of random numbers and timing the result, the following graph can be generated.



On the y-axis is the computing time needed to sort a given list, and on the x-axis is the length of list being sorted. After running a regression in excel to fit a polynomial to the time taken to sort, we can see that the time takes approximatly $n^2 + 0.003n$ to sort the list. The exact time taken doesn't match the theoretical analisys exactly, however the order is the same, both showing that the algorithm is $O(n^2)$, taking quadratic time to complete. ∎