Homework 4

Section 3.1

4. Describe an algorithm that takes as input a list of \( n \) integers and produces as output the largest difference obtained by subtracting an integer in the list from the one following it.

Solution:

```plaintext
procedure maxDiff(a_1, a_2, …, a_n: integers )
maxdiff := -∞
for i := 1 to n-1
if a_{i+1} - a_i > maxdiff then
maxdiff := a_{i+1} - a_i
end
return maxdiff
```

8. Describe an algorithm that takes as input a list of \( n \) distinct integers and finds the location of the largest even integer in the list or returns 0 if there are no even integers in the list.

Solution:

```plaintext
procedure maxEven(a_1, a_2, …, a_n: integers)
k := 0
maxeven := -∞
for i := 1 to n
if a_i % 2 == 0 and a_i > maxeven then
k := i
maxeven := a_i
end
return \{ k the desired location, otherwise 0 \}
```

34. Use the bubble sort to sort 6, 2, 3, 1, 5, 4, showing the lists obtained at each step.
Solution:

First pass: 2, 3, 1, 5, 4, 6
Second pass: 2, 1, 3, 4, 5, 6
Third pass: 1, 2, 3, 4, 5, 6
Fourth pass: 1, 2, 3, 4, 5, 6

38. Use the insertion sort to sort the list in Exercise 34, showing the lists obtained at each step.

Solution:

First pass: 6, 2, 3, 1, 5, 4
Second pass: 2, 6, 3, 1, 5, 4
Third pass: 2, 3, 6, 1, 5, 4
Fourth pass: 1, 2, 3, 6, 5, 4
Fifth pass: 1, 2, 3, 5, 6, 4
Sixth pass: 1, 2, 3, 4, 5, 6

Section 3.2

10. Show that $x^3$ is $O(x^4)$ but that $x^4$ is not $O(x^3)$.

Solution:

Since $x^3 \leq x^4$ for all $x > 1$, we know that $x^3$ is $O(x^4)$ (witnesses $C = 1$ and $k = 1$). On the other hand, if $x^4 \leq Cx^3$, then (dividing by $x^3$) $x \leq C$. Since this latter condition cannot hold for all large $x$, no matter what the value of the constant $C$, we conclude that $x^4$ is not $O(x^3)$.

26. Give a big-$O$ estimate for each of these functions. For the function $g$ in your estimate $f(x)$ is $O(g(x))$, use a simple function $g$ of smallest order.
   a) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
   b) $(2^n + n^2)(n^3 + 3^n)$

Solution:

The approach in these problems is to pick out the most rapidly growing term in each sum and discard the rest (including the multiplicative constants).

a) This is $O(n^3 \cdot \log n + \log n \cdot n^2)$, which is the same as $O(n^3 \cdot \log n)$.

b) Since $2^n$ dominates $n^2$, and $3^n$ dominates $n^3$, this is $O(2^n \cdot 3^n) = O(6^n)$. 
Section 3.3

12. Consider the following algorithm, which takes as input a sequence of $n$ integers $a_1, a_2, \ldots, a_n$ and produces as output a matrix $M = \{m_{ij}\}$ where $m_{ij}$ is the minimum term in the sequence of integers $a_i, a_{i+1}, \ldots, a_j$ for $j \geq i$ and $m_{ij} = 0$ otherwise.

initialize $M$ so that $m_{ij} = a_i$ if $j \geq i$ and $m_{ij} = 0$ otherwise

for $i := 1$ to $n$
for $j := i + 1$ to $n$
for $k := i + 1$ to $j$

$m_{ij} := \min(m_{ij}, a_k)$

return $M = \{m_{ij}\}$ \{$m_{ij}$ is the minimum term of $a_i, a_{i+1}, \ldots, a_j$\}

(a) Show that this algorithm uses $O(n^3)$ comparisons to compute the matrix $M$.

(b) Show that this algorithm uses $\Omega(n^3)$ comparisons to compute the matrix $M$. Using this fact and part (a), conclude that the algorithm uses $\Theta(n^3)$ comparisons. [Hint: Only consider the cases where $i \leq n/4$ and $j \geq 3n/4$ in the two outer loops in the algorithm.]

(a) There are three loops, each nested inside the next. The outer loop is executed $n$ times, the middle loop is executed at most $n$ times, and the inner loop is executed at most $n$ times. Therefore the number of times the one statement inside the inner loop is executed is at most $n^3$. This statement requires one comparison, so the total number of comparisons is $O(n^3)$.

(b) We follow the hint, not worrying about the fractions that might result from roundoff when dividing by 2 or 4 (these don’t affect the final answer in big-Omega terms). The outer loop is executed at least $n/4$ times, once for each value of $i$ from 1 to $n/4$ (we ignore the rest of the values of $i$). The middle loop is executed at least $n/4$ times, once for each value of $j$ from $3n/4$ to $n$. The inner loop for these values of $i$ and $j$ is executed at least $(3n/4) - (n/4) = n/2$ times. Therefore the statement within the inner loop, which requires one comparison, is executed at least $(n/4)(n/4)(n/2) = n/32$ times, which is $\Omega(n^3)$. The second statement follows by definition.