CSC 212 Homework # 1 - Solution to Selected Problems
Performance Analysis

Problem 1

1. Show that $5n^2 + 2n + 1$ is $O(n^2)$

   Answer $5n^2 + 2n + 1$ is $O(n^2)$ if $5n^2 + 2n + 1 \le cn^2$ for $c > 0$ and $n \ge n_0$, where $n_0 \ge 0$. $5n^2 + 2n + 1 \le cn^2$ then $5 + \frac{2}{n} + \frac{1}{n^2} \le c$.

   We can take $c = 10$ and $n_0 = 1$.

2. What is the Big oh of $n^2 + n\log(n)$? prove your answer.

   Answer $n^2 + n\log(n)$ is $O(n^2)$.

   $n^2 + n\log(n)$ is $O(n^2)$ if $n^2 + n\log(n) \le cn^2$ for $c > 0$ and $n \ge n_0$, where $n_0 \ge 0$.

   $n^2 + n\log(n) \le cn^2$ then $1 + \frac{1}{n}\log(n) \le c$.

   We can take $c = 2$ and $n_0 = 2$.

3. Show that $2n^3 \notin O(n^2)$.

   Answer if $2n^3$ is $O(n^2)$ then $2n^3 \le cn^2$.

   We obtain then $2n \le c$.

   Whatever the positive value that we choose for c, we can always find a value for n_0 such that the inequality above is not valid for $n \ge n_0$.

4. Assume that the expression below gives the processing time $f(n)$ spent by an algorithm for solving a problem of size n.

   $$10n + 0.1n^2$$

   (a) Select the dominant term(s) having the steepest increase in n. **Answer**: n^2

   (b) Specify the lowest Big-Oh complexity of the algorithm. **Answer**: $O(n^2)$

5. Determine whether each statement is *true* or *false* and correct the expression in the latter case:

   (a) $100n^3 + 8n^2 + 5n$ is $O(n^4)$. **True**: $O(n^3)$ is also $O(n^4)$.

   (b) $100n^3 + 8n^2 + 5n$ is $O(n^2 \log n)$. **False**: $O(n^3)$.

   Remark: If you are asked to provide the big O notation, always choose the smallest class of functions. Example: $100n^3 + 8n^2 + 5n$ is $O(n^3)$ (we do not use $O(n^4)$ even though $O(n^3)$ is also $O(n^4)$).

6. Show that $\log_a(n) \in O(\log_b(n))$ for all $a, b > 0$.

   Answer if $\log_a(n) \in O(\log_b(n))$ for all $a, b > 0$ then $\log_a(n) \le c\log_b(n)$.

   $\log_a(n) \le c\log_b(n) \Leftrightarrow \frac{\log(n)}{\log(a)} \le c\frac{\log(n)}{\log(b)}$

   If we take $c = \frac{\log(b)}{\log(a)}$, the inequlity above remains valid for any value of $n > 0$.

7. Show that $a^n \notin O(b^n)$ if $a > b > 0$.

   Answer if $a^n \in O(b^n)$ for $a > b > 0$ then $a^n \leq cb^n$.

   We obtain then $\frac{a^n}{b^n} \leq c \Leftrightarrow (\frac{a}{b})^n \leq c$.

   Since $\frac{a}{b} > 1$, whatever the positive value that we choose for c, we can always find a value for n_0 such that the inequality above is not valid for $n \geq n_0$.

. . . .

Problem 2

Analyze the following code excerpts:

1.
```
1 int sum = 0;
2 for (int i = 1; i <= n ; i++)
3         for (int j = 0; j <  2* i ; j++)
4                 sum += j;
5 return sum;
```

Line	Frequency
1	1
2	n+1
3	$n^2 + 2n$
4	$n^2 + n$
5	1
Big O	n^2

```
1 for (int i = 0; i < n * n * n; i++) {
2         System.out.println(i);
3         for (int j = 2; j < n; j++)
4                 System.out.println(j); }
5  System.out.println("End!");
```

Line	Frequency
1	$n^3 + 1$
2	n^3
3	$n^4 - n^3$
4	$n^4 - 2n^3$
5	1
Big O	n^4

3.
```
1 int k = 100, sum = 0;
2 for (int i = 0; i < n; i++)
3         for (j = 1; j <= k; j++) {
4                 sum = i + j;
5                 System.out.println(sum);
6         }
```

Line	Frequency
1	1
2	$n+1$
3	$101n$
4	$100n$
5	$100n$
Big O	n

. . . .

Problem 3

1. Given an n-element array X, Algorithm B chooses $\log n$ elements in X at random and executes an $O(n)$-time calculation for each. What is the worst-case running time of Algorithm B? (Question R-4.30 page 184 of the textbook).

   Answer:$O(n \log n)$

2. Given an n-element array X of integers, Algorithm C executes an $O(n)$-time computation for each even number in X, and an $O(\log n)$-time computation for each odd number in X. What are the best-case and worst-case running times of Algorithm C? (Question R-4.31 page 184 of the textbook).

   Answer: Best case: all elements are odd, the running time is $O(n \log n)$. Worst case: all elements are even, the running time is $O(n^2)$.

3. Give in asymptotic notation the running time for the following algorithms:

   (a) Vector-vector addition (the vectors are of size n). **Answer:** $O(n)$

   (b) Dot product of two vectors (the vectors are of size n). **Answer:** $O(n)$

   (c) Matrix-vector multiplication (the matrix is of size $m \times n$, the vector is of size n). **Answer:** $O(nm)$

   (d) Matrix addition (the two matrices are of size $m \times n$). **Answer:** $O(nm)$

   (e) Matrix-Matrix multiplication (the two matrices are of size $m \times k$ and $k \times n$ respectively). **Answer:** $O(nmk)$

. . . .