1 Time analysis

What is the running time of the following algorithm (using big-O notation)? Show your analysis in two ways: (1) a detailed analysis which counts the number of times each line of code executes, (2) using simpler analysis with big-O notation.

1: \textit{result} \leftarrow 1
2: \textbf{for} \ i = 1 \textbf{to} \ n \ \textbf{do}
3: \textbf{for} \ j = 1 \textbf{to} \ i \ \textbf{do}
4: \textit{result} \leftarrow \textit{result} + 1
5: \textbf{end for}
6: \textbf{end for}

2 Asymptotic Notations

For this problem, you need to base your answer on the definitions of the asymptotic notations (not the common sense rules).

1. Show $8n^3 \log n + 14n^2 = \Theta(n^3 \log n)$.

2. If $f(n) = O(g(n))$, can we conclude $2f(n) = O(2g(n))$? Justify your answer.

3. Prove for any integer constant $a$ and real constant $b$, $(n + b)^a = \Theta(n^a)$. In fact, this holds even when $a$ is a real constant, but you do not have to prove it.

3 Asymptotic order

The following functions are selected from problem 3-3(a) on page 61 of your textbook (p.58 if you use the 2nd edition). Order these functions asymptotically as explained in the textbook. You do not need to give proof, but you may want to work out yourself for self-study. Note that \textit{lnn} means $\log_2(n)$.

$n^2, n!, (\frac{3}{2})^n, n^3, \log^2n, n \cdot 2^n, \ln n, 1, 2^{\log n}, e^n, 2^n, n \log n, n$.

4 Finding common items in two sorted lists

You are given two \textit{sorted} lists $A$ and $B$ of items (say numbers). The lengths of $A$ and $B$ are both $n$. Now design an $O(n)$ time algorithm to find all distinct items that are in both $A$ and $B$. Note: you need to first describe your high-level idea and then write down the algorithm in pseudo-code (but without too much coding details), and finally justify your algorithm and provide running time analysis.

5 Sum of two numbers

You are given an array $A[1 \ldots n]$ of arbitrary integers and another integer $u$. The problem is to check if the array $A$ has two elements $x$ and $y$ such that $x + y = u$. Present an algorithm to solve this problem. What is the run time of your algorithm?