

COMP 4030/6030: Assignment 4
Due date: 03/21/2016

Email programming solutions to the TA (Quang Tran, qmtran@memphis.edu). Put “COMP 4030/6030 assignment 3” on the subject line.

1. Programming exercise: **maxsum** computes the largest possible value of summing consecutive numbers from an input list of numbers. For example, given the input $[10, -20, 5, -3, 7, 8, -10, 4]$, the output is 17. This is because the largest possible sum of consecutive numbers from this list is 17 ($5 - 3 + 7 + 8$). Now, write a Python program to implement **maxsum** based on each of the following strategies (each strategy corresponds to a different Python implementation):
 - (a) (15 points) Look at all possible intervals of consecutive numbers in the input list. Compute and remember the maximum sum. This function should run in $O(n^3)$ time.
 - (b) (15 points) Modify the previous strategy so that your implementation only takes $O(n^2)$ running time. Of course, if your previous strategy is already $O(n^2)$, you get points for this automatically.
 - (c) (15 points) Divide and conquer: divide the list in two halves. Compute max sum for the left and right halves separately. Then, compute max sum for the case when the interval can possibly cross the two halves. This can be done in $O(n)$ time. If done right, this strategy only takes $O(n \log n)$ running time.
2. (15 points) Use the arithmetic sum to find out the value of $1 + 2 + 3 + 4 + \dots + 500$.
3. (15 points) Use the geometric sum to find out the value of $1 + 3 + 3^2 + 3^3 + \dots + 3^{20}$.
4. (15 points) Use substitution to find the complexity of $T(n) = 5n + T(n - 1)$. (Assume $T(1) = 1$).
5. (15 points) Use substitution to find the complexity of $T(n) = n^2 + 4 \cdot T(\frac{n}{2})$. (Assume $T(1) = 1$).
6. (15 points) Use substitution to find the complexity of $T(n) = 4n + T(\frac{n}{2})$. (Assume $T(1) = 1$).