

CS 301-01: Algorithm Design and Analysis

HW 1 (Given September 8, 2017; Due September 15, 2017)

Each (sub-)question is worth 10 points unless otherwise stated

1. Given an array A consisting of $A[1], A[2], \dots, A[n]$, you would like to output a two-dimensional $n \times n$ array B such that $B[i, j], \text{ for } (i < j) = \sum_{k=i}^{k=j} A[k]$. When $i \geq j$, $B[i, j]$ is unspecified (can be anything, we don't care).
 - (a) For some function f , given a bound of the form $O(f(n))$ on the running time of this algorithm
 - (b) Show that the running time is also $\Omega(f(n))$. What does it imply?
2. Solve the recurrence (assume N is a power of 2),

$$C_N = C_{N/2} + N^2 \quad , \quad N \geq 2, C_1 = 0 \quad (1)$$

3. Design a recursive program that converts a string of numbers into its numeric equivalent i.e. "12345" is converted into a number 12,345.
 - (a) What is the worst case time complexity of your algorithm
 - (b) Implement the above algorithm. Plot the theoretical estimate from (a) above and the actual running time from your implementation for multiple values of n .