Problem 1. Assume you are given arrays, \( A \) and \( B \), of lengths \( m \) and \( n \), respectively. Write pseudo-code to compute the following sum, \( S \),

\[
S = \sum_{i=1}^{m} \sum_{j=1}^{n} \left( A[i] \cdot B[j] \cdot 10^{i+j} \right)
\]

Problem 2. Given the runtime of insertion sort algorithm to be \( 2n^2 \) and that of Merge sort algorithm to be \( 50n \log_2 n \) running on the same machine. Show the following:

1. On a single plot show the growth curves for the two algorithms as \( n \) grows from 1 to 1000.
2. On your plot, show the value for \( n \) after which a runtime crossover occurs.
3. For which values of \( n \) would you prefer each algorithm?

Problem 3. Assume that your machine takes time(work), \( i + j \) to compare two elements in locations \( i \) and \( j \). Analyze how much exact comparison time Bubble sort takes in the worst case to sort an array of size \( n \). Assume that the array is stored in locations 1 to \( n \). You may assume regular bubble sort algorithm as discussed in the class.