1) (6 pts) Consider a text comprising of 500 ones. Determine the number of comparisons encountered to search for the following patterns in this text. Show all the work.

111111111111111111111111111111111111111111 (500 ones)

(a) 11101  (b) 01111

(a) 11101
   It would take \(2\) comparisons per block to decide if the pattern does not match for the block.
   \# blocks = \((500 - 5 + 1)\) = 496.
   total \# comparisons = 496 \times 4 = 1984.

(b) 01111
   It would take just \(1\) comparison per block to decide if the pattern does not match for the block.
   \# blocks = \((500 - 5 + 1)\) = 496.
   total \# comparisons = 496 \times 1 = 496.

2) (7 pts) Use a \(\Theta(n)\) algorithm to determine the number of substrings that start with a 'C' and end with a 'D' in the string ACDABCADACD. Show all the work.

\[
\begin{array}{cccccccc}
\# Cs & 0 & 1 & 1 & 1 & 2 & 3 & 3 & 3 & 3 & 4 & 4 \\
\# substrings & 0 & 0 & 1 & 1 & 1 & 1 & 4 & 4 & 4 & 8 & \end{array}
\]
3) (12 pts) Consider the following pseudo code for insertion sort. Derive the expressions for the best-case and worst-case number of comparisons and the overall time-complexity of the algorithm with respect to an appropriate asymptotic notation.

**Algorithm** 

InsertionSort(A[0..n-1]):

//Sorts a given array by insertion sort
//Input: An array A[0..n-1] of n orderable elements
//Output: Array A[0..n-1] sorted in nondecreasing order

for \( i = 1 \) to \( n-1 \) do

\[ v = A[i] \]

\( j = i - 1 \)

while \( j \geq 0 \) and \( A[j] > v \) do


\( j = j - 1 \)

\[ A[j+1] = v \]

---

**Best-case**

\[ \sum_{i=1}^{n-1} (i-1) = \frac{n(n-1)}{2} = n - 1 \]

**Worst-case**

\[ \sum_{i=1}^{n-1} 1 = \frac{n(n-1)}{2} = 1 \]

\[ \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \]

\[ \sum_{i=1}^{n-1} [i-1 - 0 + 1] = \frac{n(n-1)}{2} \]

**Overall**

\[ \lim_{n \to \infty} \frac{\text{best-case}}{\text{worst-case}} = \lim_{n \to \infty} \frac{\frac{n(n-1)}{2}}{\frac{n(n-1)}{2}} = \lim_{n \to \infty} \frac{2}{n} = 0 \]

Overall time complexity = \( O\left(\frac{n(n-1)}{2}\right) = O(n^2) \)