CS 2210 Assignment 1

Question 4:

(a) Prove that this algorithm is correct by providing the following:

i. (2 marks) Show that the algorithms terminates.

The algorithm will terminate because n it will only run until = n – 1 and n is finite. Every time the inner for-loop runs, the variable j will increment up to a maximum of n-1. Then, the outer loop will execute and the variable i will be incremented by 1, also to a maximum of n-1.

1. (2 marks) Show that the algorithm always produces the correct answer.

The algorithm will produce the correct answer because it recursively calls the function isPalindrome() on every possible substring of the original input and then increments c if the substring is a palindrome.

1. (4 marks) Compute the time complexity of this algorithm in the worst case. You must explain how you computed the time complexity, and you must give the order of the complexity. Assume that the function isPalindrome() takes only a constant time to execute.

The assignment c = 0 at the start and the return c each take up one unit of time. At the start of the outer loop, the assignment i <- 0 and the operation n – 1 take up a total of 2 units of time per loop.

The inner loop executes n times for every iteration of the outer loop. Since the outer loop executes n times, that means the inner loop will execute n^2 times. At the beginning of the inner loop, there are 2 primitive operations j <- i and n = 0 that take up 2 units of time total and are executed each time the inner loop runs, meaning these will be executed n^2 times.

Within the inner loop, there are 3 operations that will take up 1 unit of time each: the assignment statement “substring ← T [i to j];”, the recursive call for isPalindrome(), and the incrementation of c. Since we are discussing the worst case, the line that increments c will always execute.

The time complexity is therefore The order of the complexity is O()

CS 2210 ASSIGNMENT

5. Consider the following algorithm that updates the values in the array A.

(a)  (3 marks) Compute the time complexity of this algorithm in the best case. You must explain how you computed the time complexity, and you must give the order of the complexity.

In the best case, all integers are positive or 0, so the else-if statement won’t execute.

The line i 🡸 0 uses 1 unit of time.

The while loop executes n times. The comparison operation i < n uses 1 unit of time per loop, or n units. The if i = n statement uses 1 unit of time per loop, and the code within will only execute once in the best case, adding 1 unit of time. The else-if statement will be skipped altogether in the best case, bringing us to the last line, which uses 2 operations per loop, or 2n units.

The time complexity is , and the order of the complexity is O(n).

(b)  (3 marks) Compute the time complexity of this algorithm in the worst case. You must explain how you computed the time complexity, and you must give the order of the complexity.

In the worst case, all elements of the array are positive.

The algorithm starts with I 🡸 0, taking up 1 unit of time.

Since the algorithm backtracks to index 0 every time it encounters a positive element, the while-loop will execute times. Since everything inside the while-loop can be considered a constant, we can say that the complexity is ( and it is O).