150015-Semester II - 5781

Data Structures I

**Homework Assignment #5**

**Question 1**

Write a recursive function that accepts an integer n and returns the sum of all the divisors of n.

For example: If n=12 then the returned vale will be 28 (1+2+3+4+6+12)

Note: you may solve this problem using two functions, one non-recursive that calls the recursive one.

**Question 2**

Write a reursive function that computes ⎣log3n⎦ for any natural number n >= 1. You are not allowed to use and log or exponent functions.

**Question 3**

Given: An empty queue Q and the following function

func(n)

if n > 0

enqueue(Q, n)

if n%2 == 0

enqueue(Q, n/2)

print(dequeue(Q))

func(n-1)

Q is global and any change is also known in the recursive calls.

1. What will be the output for the following call to the function, func(7)?
2. What will be the content of Q after the above call to the function?

**Question 4**

Given: Two stacks containing natural numbers. Write a recursive function that returns true if the stacks are congurent, otherwise, returns false.

Two stacks are congurent if all their elements are the same in value and in the same place.

The stacks are global and you do not need to accept them, any changes you make in them will be seen in all the recursive calls.

The solution must use the standard stack interface functions such as push and pop etc’.

At the end of the solution the stacks must be without any changes, that is just like they were before the call.

You may not use loops such as for, do, while etc and you can not use any other data structures.

**Question 5**

a. What is the worst case run time complexity of the following algorithm, comprised of 3 commands, to sort an array A[1,….,n]

1) Perform Mergesort on A[1,…, ⎣n/2⎦]

2) Perform QuicksortA[⎣n/2⎦+1,…,n]

3) Merge A[1,…, ⎣n/2⎦] with A[(⎣n/2⎦)+1,…,n]

b. What is the average case run time complexity of the above algorithm?