

CSE101 – Spring 2020

Programming Assignment #5

(10 points, Submission due date: 6 May 2020)

Instructions

For each of the following problems, create an error-free Python program.

- Each program should be submitted in a separate Python file that follows a particular naming convention: Submit the answer for problem 1 as “Assign5Answer1.py” and for problem 2 as “Assign5Answer2.py” and so on.
- These programs should execute properly in PyCharm using the setup we created in lab.
- At the top of every file add your name and Stony Brook email address in a comment.
- Write 2 test cases to test your code for all problems that do not take user input.

Problems

Problem 1: Answer the following questions.

(2 points)

For the following, copy the questions into a text file and write your answers for each question. Name the text file Assign5Answer1.

- I. Place the following algorithm time complexities in order from fastest (least number of comparisons) to the slowest: $n \log n$, n , n^2 , $\log n$, $2n$.
- II. In your own words, explain the two characteristics that a recursive solution must have.
- III. Why are divide-and-conquer algorithms often very efficient in terms of time complexity?

Problem 2: Recursive functions for numbers

(3 points)

For this problem you must write the functions in a recursive manner (i.e. the function must call itself) – it is not acceptable to submit an iterative solution to these problems.

A. Complete the following function that uses recursion to concatenate the numbers in forward order to form a string.

```
# If n = 9, it returns '0123456789'
# If n = 13, it returns '012345678910111213'
# Pre-condition: n >= 0
#
def concat_to(n):
    return None # Replace this with your implementation
```

B. Complete the following function that uses recursion to concatenate the numbers in reverse order to form a string.

```
# If n = 9, it returns '9876543210'
# If n = 13, it returns '1312111109876543210'
# Pre-condition: n >= 0
#
def concat_reverse_to(n):
    return None # Replace this with your implementation
```

C. Complete the recursive function `gcd(m, n)` that calculate the greatest common denominator of two numbers with the following rules:

```
# If m = n, it returns n
# If m < n, it returns gcd(m, n-m)
# If m > n, it returns gcd(m-n, n)
#
def gcd(m,n):
    return None # Replace this with your implementation
```

Problem 3: Recursive functions for lists

(5 points)

For this problem you must write the functions in a recursive manner (i.e. the function must call itself) – it is not acceptable to submit an iterative solution to these problems.

A. Complete the following function that uses recursion to find and return the even elements in the list `u`.

```
# find_evens([1, 2, 3, 4] returns [2, 4]
# find_evens([1, 2, 3, 4, 5, 6, 7, 8, 9, 10] returns [2, 4, 6, 8, 10]
#
def find_evens(u):
    return None # Replace this with your implementation
```

B. Complete the following recursive function that returns the zip of two lists `u` and `v` of the same length. Zipping the lists should place the first element from each into a new array, followed by the second elements, and so on (see example output).

```
# zip([1, 2, 3], [4, 5, 6]) returns [1, 4, 2, 5, 3, 6]
#
def zip(u, v):
    return None # Replace this with your implementation
```

C. Complete the following recursive function that removes all occurrences of the number `x` from the list `nums`.

```
# remove_number(5, [1, 2, 3, 4, 5, 6, 5, 2, 1]) returns [1, 2, 3, 4, 6, 2, 1]
#
def remove_number(x, nums):
    return None # Replace this with your implementation
```