National University of Singapore
School of Computing
CS1010S: Programming Methodology

Extra Practice 3

Help

Yes, we put these functions here so that you can refer to them easily.

```
def sum(term, a, next, b):
    if a > b:
        return 0
    else:
        return term(a) + sum(term, next(a), next, b)

def fold(op, f, n):
    if n == 0:
        return f(0)
    else:
        return op(f(n), fold(op, f, n-1))
```

Order of Growth

Determine the time and space complexity of all these functions.

(a)
```
def lol1(n, m):
    result = 0
    for i in range(n):
        for j in range(m):
            result += 1
    return result
```

(b)
```
def lol2(n):
    result = 0
    for i in range(n):
        for j in range(n):
            result += 1
    return result
```

(c)
```
def lol3(n):
    result = ''
    for i in range(n):
        result += 'a'
    return result
```

(d)
```
def lol4(n):
    if n == 0:
        return 0
    else:
        return lol4(n - 1)
```

(e)
```
def lol5(n):
    result = 0
    for i in range(n):
        for j in range(i, n):
            result += 1
    return result
```

(f)
```
def lol6(n):
    if n >= 1:
        return 0
    print("CS1010S is fun!")
    lol6(n // 2)
    lol6(n // 2)
```

(g)
```
def lol7(n):
    for i in range(n):
        for j in range(n + 1, i):
            print("Hello, I am Baymax")
```

(h)
```
def lol8(n):
    if n < 2:
        print("Less than two")
        return 1
    else:
        for j in range(1,n):
            print("CS1010S is fun!")
        a = lol8(n // 3)
        b = lol8(n // 3)
        c = lol8(n // 3)
        return a + b + c
```

(i)
```
def lol9(n):
    if n <= 1:
        return
    print("CS1010S")
    for i in range(1, 2):
        lol9(n - 1)
```

Higher Order Functions

(a) Define a function **`total`** that produces the output of the following code using either **`sum`** or **`fold`**.

```
2 + 4 + 6 + 8 + 10
```

(b) I would like to convert a password such as "orange" into a string that comprises **only** of "*", depending on how long my word is. This function will be named **`convert`** and take in a word string as an input while returning the converted word. You may assume that the word will be at least one letter long.
Sample Output:

```
>>> convert("orange")
'******'
>>> convert("ap13")
'****'
```

- Use an iterative approach to solve this. What is the time and space complexity?
- Use a recursive approach to solve this. What is the time and space complexity?
- Use the **`fold`** function to solve this.
- Explain if the **`sum`** function can be used to solve this. If not, explain what change needs to be made to the original function and define it in terms of **`sum`**.

(c) Now, I would like to filter out the letters "o" and "a" because I don't really like them. Define a function **`remove`** that takes in a word and returns the new word with all the "o"s and "a"s removed.

```
>>> remove("orange")
'rnge'
>>> remove("oooaaat")
't'
```

- Use an iterative approach to solve this. What is the time and space complexity?
- Use a recursive approach to solve this. What is the time and space complexity?
- Use the **`fold`** function to solve this.
- Explain if the **`sum`** function can be used to solve this. If not, explain what change needs to be made to the original function and define it in terms of **`sum`**.
- ***How do you modify the functions to remove all vowels in a word?***