Recursive Algorithm

Peasant Method

peasantAlgorithm.py

```python
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 16 20:39:06 2020

@author: farshad
@email : fzcomputerscience@gmail.com
"""

def peasantMultiplication(x,y):
    if(x==0):
        return 0
    x1 = x/2
    y1 = y + y
    result = peasantMultiplication(x1,y1)
    if(x % 2 == 0):
        return result
    else:
        return result + y


result = peasantMultiplication(2,3)
print(result)

```

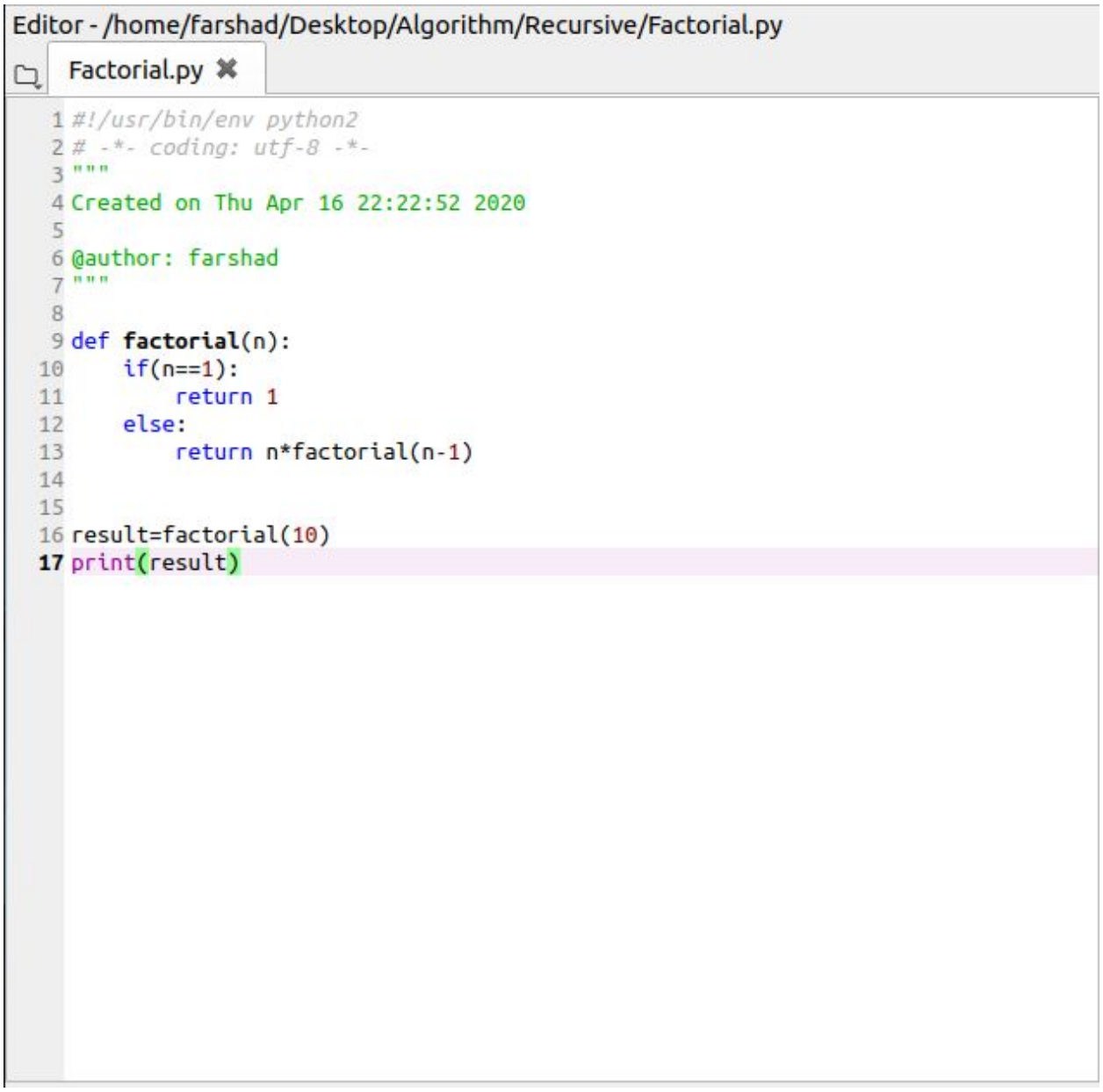
```
def peasantMultiplication(x,y):
        if(x==0):
        return 0
        x1 = x/2
        y1 = y + y
        result = peasantMultiplication(x1,y1)
        if(x % 2 == 0):
        return result
        else:
        return result + y


result = peasantMultiplication(2,3)
print(result)
```

Lattice Algorithm

- Not Ready Yet

Factorial of a Number




```
def factorial(n):
        if(n==1):
        return 1
        else:
        return n*factorial(n-1)


result=factorial(10)
print(result)
```

Merge Sort

Editor - /home/farshad/Desktop/Algorithm/Recursive/MergeSort.py

MergeSort.py

```
"""
Created on Sun Apr 26 16:59:08 2020
@author: farshad
@email: fzcomputerscience@gmail.com
"""
def mergeSort(inputArray):
    length = len(inputArray)
    if(length<2):
        return
    else:
        middle = length/2
        subInputArrayLeft = inputArray[0:middle]
        subInputArrayRight = inputArray[middle:length]
        mergeSort(subInputArrayLeft)
        mergeSort(subInputArrayRight)
        merge(inputArray,subInputArrayLeft,subInputArrayRight,middle)

def merge(inputArray,subInputArrayLeft,subInputArrayRight,middle):
    leftIndex = 0
    rightIndex = 0
    for i in range(len(inputArray)):
        if(leftIndex>=middle):
            inputArray[i] = subInputArrayRight[rightIndex]
            rightIndex = rightIndex + 1
        elif((rightIndex+middle)>=len(inputArray)):
           inputArray[i] = subInputArrayLeft[leftIndex]
           leftIndex = leftIndex + 1
        elif(subInputArrayLeft[leftIndex]<=subInputArrayRight[rightIndex]):
            inputArray[i] = subInputArrayLeft[leftIndex]
            leftIndex = leftIndex + 1
        elif(subInputArrayLeft[leftIndex]>subInputArrayRight[rightIndex]):
            inputArray[i] = subInputArrayRight[rightIndex]
            rightIndex = rightIndex + 1

numbers = [200,201,100,10,7,6,5,3,7,1,0,-1,0,-2,0,100,13]
mergeSort(numbers)
for i in range(len(numbers)):
    print(numbers[i])
```

```
def mergeSort(inputArray):
	length = len(inputArray)
	if(length<2):
	return
	else:
	middle = length/2
	subInputArrayLeft = inputArray[0:middle]
	subInputArrayRight = inputArray[middle:length]
	mergeSort(subInputArrayLeft)
	mergeSort(subInputArrayRight)
	merge(inputArray,subInputArrayLeft,subInputArrayRight,middle)

def merge(inputArray,subInputArrayLeft,subInputArrayRight,middle):
	leftIndex = 0
	rightIndex = 0
	for i in range(len(inputArray)):
	if(leftIndex>=middle):
	inputArray[i] = subInputArrayRight[rightIndex]
	rightIndex = rightIndex + 1
	elif((rightIndex+middle)>=len(inputArray)):
	inputArray[i] = subInputArrayLeft[leftIndex]
	leftIndex = leftIndex + 1
	elif(subInputArrayLeft[leftIndex]<=subInputArrayRight[rightIndex]):
	inputArray[i] = subInputArrayLeft[leftIndex]
	leftIndex = leftIndex + 1
	elif(subInputArrayLeft[leftIndex]>subInputArrayRight[rightIndex]):
	inputArray[i] = subInputArrayRight[rightIndex]
	rightIndex = rightIndex + 1

numbers = [200,201,100,10,7,6,5,3,7,1,0,-1,0,-2,0,100,13]
mergeSort(numbers)
for i in range(len(numbers)):
	print(numbers[i])
```

References

Erickson, J. (2019). *Algorithms*. Etats-Unis: Jeff Erickson.