</a>

Recursive Algorithms

Notebook description

This notebook stores the work as part of the Introduction to Discrete Mathematics for Computer Science Specialization. Course URL: https://www.coursera.org/specializations/discrete-mathematics (https://www.coursera.org/specializations/discrete-mathematics). Code included in this repository is written by myself, Kristen Phan. Below are recursive algorithms written to solve various math problems.

Assigments

Assignment 1

Assigment description: Develop a Python method change(amount) that for any integer amount in the range from 24 to 1000 returns a list consisting of numbers 5 and 7 only, such that their sum is equal to amount. For example, change(28) may return [7, 7, 7, 7], while change(49) may return [7, 7, 7, 7, 7, 7, 7] or [5, 5, 5, 5, 5, 5, 5, 7, 7] or [7, 5, 5, 5, 5, 5, 5, 5, 7].

In [5]:

```
# create a custom function called 'change'
def change(amount):
  if amount == 24:
    return [5, 5, 7, 7]
  if amount == 25:
    return [5, 5, 5, 5, 5]
  if amount == 26:
    return [5, 7, 7, 7]
  if amount == 27:
    return [5, 5, 5, 5, 7]
  if amount == 28:
    return [7, 7, 7, 7]
  coins = change(amount - 3)
  coins.append(3)
  return coins
```

In [6]:

```
# test function 'change' with the amount of 100
change(100)
```

Out[6]:

```
[7,
 7,
 7,
 7,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3]
```

Assignment 2

Assignment description: The number of moves required to solve the Hanoi Towers puzzle for n=1n=1 and n=2n=2 discs is equal to 1 and 3, respectively. Implement the recursive solution for the Hanoi Towers (described in the lectures) and count the number of moves for n=6n=6 discs.

In [7]:

```
# create a custom function called hanoi count the number of moves required
def hanoi(n, src, temp, dest):
    if n == 1:
        return 1

    return hanoi(n - 1, src, dest, temp) + hanoi(1, src, temp, dest) + hanoi(n
- 1, temp, src, dest)
```

In [8]:

```
# test function 'hanoi' for the below parameters
hanoi(6, src=1, temp=2, dest=3)
```

Out[8]:

```
63
```