CS100 Tutorial 12

Hands-on small projects

Task 1: Climb the stairs

There are n stairs and suppose you are at the 0 stair. You can climb at most 3 stairs (i.e., 1 or 2 or 3 stairs) at each step, the problem is to compute the number of possible solutions to reach the nth stair.

1. Implement a recursive function to solve this problem
2. Implement a non-recursive function to solve this problem
3. Compare the performance of these two alternative implementations (Hint: use time module to compare execution time for different values of n)

Task 1: Climb the stairs

Testcases:

```
climbStairs(3)
>>>4
climbStairs(4)
>>>7
```

Task 2: Monty Hall problem

The Monty Hall problem is a brain teaser, in the form of a probability puzzle, loosely based on the American television game show Let's Make a Deal and named after its original host, Monty Hall. The problem was originally posed (and solved) in a letter by Steve Selvin to the American Statistician in 1975. It became famous as a question from a reader's letter quoted in Marilyn vos Savant's "Ask Marilyn" column in Parade magazine in 1990 (vos Savant 1990a):

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?

Vos Savant's response was that the contestant should switch to the other door. Under the standard assumptions, contestants who switch have a 2 / 3 chance of winning the car, while contestants who stick to their initial choice have only a 1 / 3 chance.

Task 2: Monty Hall problem

Implement a program to mimic the game.

Hints:

1. Use dict to model the doors 1,2,3 and things behind the doors
2. Randomly add car behind one door, while goats are behind other doors using random module
3. Use try statement to handle exceptions for invalid inputs so that the game can continue

Task 2: Monty Hall problem

Testcases:

```
=============================
Choose a door to open:1
"goat" behind the door 0
Switch to 2?(y/n)n
You Win.
Do you want to try once more?(y/n)y
=============================
Choose a door to open:1
"goat" behind the door 2
Switch to 0?(y/n)n
I Win!
Do you want to try once more?(y/n)n
>>>
```

Task 3: n-D Vector

Implement a n-dimension vector (for any n>=1) class

The class should supports:

1. Add (+) and Sub (-) between vectors (with same dimensions)
2. Multi(*) and Div(/) between a vector and a scalar
3. Length of vector which is Euclidean norm of the vector (using attribute, i.e., @property decorate)
4. Pretty print of the vector

Task 3: n-D Vector

Testcases:

```
>>> v1 = Vector(3, 4, 5)
>>> v2 = Vector(5, 6, 7)
>>> print(v1)
Vector({3},{4},{5})
>>> print(v1+v2)
Vector({8},{10},{12})
>>> print(v1-v2)
Vector({-2},{-2},{-2})
>>> print(v1*3)
Vector({9},{12},{15})
>>> print(v1/2)
Vector({1.5},{2.0},{2.5})
>>> print(v1.length)
7.0710678118654755
```