Homework 1 Part 2

This is an individual assignment.

In [2]:

```
import random
import math
```

Description

Create or edit this Jupyter Notebook to answer the questions below. Use simulations to answer these questions. An analytical solution can be useful to check if your simulation is correct but analytical solutions alone will not be accepted as a solution to a problem.

Problem 1

Consider repeatedly rolling a fair 6-sided die.

1. Create a simulation to compute the probability that the top face will be 4 at least once on 10 rolls of the die?
2. Create a simulation to compute the probability that the top face will be 4 at least once on 20 rolls of the die?
3. Create a simulation to compute how many rolls of the die would you have to do to be 90% confident that you would see at least one 4?
4. Using the formula you have computed in problem 2 part 4, make a Python function that takes in the target value $p$ and outputs the required number of rolls of an integer.
   A. Find the values for $p=0.95$ and $p=0.99$.
   B. Use your simulation to verify that the number of rolls you specified is sufficient to achieve $p\geq 0.95$.

In [3]:

```
# for both 1 and 2
def roll_dies(num_rolls, num_trials):
    out = [random.choices(range(1, 7), k=num_rolls) for _ in range(int(num_trials))]
    num_success = [4 in element for element in out].count(True)
    return num_success / num_trials

p1_1 = roll_dies(num_rolls=10, num_trials=1e5)
p1_2 = roll_dies(num_rolls=20, num_trials=1e5)

print(f"p1_1: prob = {p1_1}")
print(f"p1_2: prob = {p1_2}")
```

```
p1_1: prob = 0.83876
p1_2: prob = 0.97398
```

In [11]:

```
# problem 3
def p1_3(prob, num_trials=int(1e5)):
    p_witnessed = 0
    num_rolls = 0
    while (p_witnessed < prob):
        num_rolls += 1
        p_witnessed = roll_dies(num_rolls, num_trials)
    return num_rolls

p1_3(0.9)
```

Out[11]:

```
13
```

In [12]:

```
def p1_4(prob):
    return math.ceil(math.log(1 - prob) / math.log(5/6))

print(f"prob=0.95 analytical: {p1_4(0.95)}, simulation: {p1_3(0.95)}")
```

```
prob=0.95 analytical: 17, simulation: 17
prob=0.99 analytical: 26, simulation: 26
```

In [ ]:

Problem 2

Create a simulation function where you will roll a fair 6-sided die twice. Use simulation to find out the probability of getting a even number on the first toss and a number greater than or equals 3 on the second toss.

In [13]:

```
def roll_twice():
    return random.choices(range(1, 7), k=2)

def probE(num_trials=1e5):
    experiments = [tuple(roll_twice()) for _ in range(int(num_trials))]
    success = [True if exp[0] in [2, 4,   6] and exp[1] in [3,4,5,6] else False for exp
    return float(success.count(True)) / len(experiments)

probE()
```

Out[13]:

```
0.33357
```

Problem 3

Suppose that you have a bag with 3 coins. One of them is a fair coin, but the others are biased trick coins. When flipped, the three coins come up heads with probability $\frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{4}$, respectively.

Consider the experiment where you pick one coin at random and flip it three times. Let H_i be the event that the coin comes up heads on flip i. What is the probability of the outcome $H_1\cap H_2\cap {H^c_3}$?

With small modification in your code, find out the probability of the outcome $H^c_1\cap {H^c_2} \cap {H_3}$.

Use simulation to find out the probability.

In [14]:

```
def p3(goal = ['H', 'H', 'T'], num_sim = 100_000):
    p_set = [1/2,1/5,1/4]
    pass_count = 0
    for i in range(num_sim):
        p = random.choice(p_set)
        outcome = random.choices(['H','T'], [p,1-p], k=3)
        if (outcome == goal):
            pass_count += 1
    return pass_count/num_sim

print(f"(H, H, T) : {p3(['H', 'H', 'T'])}")
print(f"(T, T, H) : {p3(['T', 'T', 'H'])}")
```

```
(H, H, T) : 0.06949
(T, T, H) : 0.13224
```

Submit Your Solutions

Along with the Notebook, include a PDF of the notebook with your solutions.