

```

import random
from tabulate import tabulate

# global parameters
p_i = 0.10 # probability of injury
p_p = 0.05 # probability of probing
p_r = 1.00 # probability of retaliation

v = 60 # payoff of winning
c = 100 # cost of serious injury
s = 2 # cost of scratch
u_init = 20 # payoff of saving time and energy (counts down every round)
u_minus = 0.5

num_trials = 100000

class Hawk:
    def __init__(self, name):
        self.name = name
        self.payoff = 0
        self.currentAction = "C"
        self.injured = False
        self.attacked = False

    def move(self):
        if self.injured:
            self.currentAction = "R"
        else:
            self.currentAction = "D"
        # print(self.name + ": " + self.currentAction)

class Mouse:
    def __init__(self, name):
        self.name = name
        self.payoff = 0
        self.currentAction = "C"
        self.injured = False
        self.attacked = False

    def move(self, u):
        if self.attacked or u == 0:
            self.currentAction = "R"
        else:
            self.currentAction = "C"
        # print(self.name + ": " + self.currentAction)

class Retaliator:
    def __init__(self, name):
        self.name = name
        self.payoff = 0
        self.currentAction = "C"
        self.injured = False
        self.attacked = False

    def move(self, opponent, u):
        if self.injured or u == 0:

```

```

        self.currentAction = "R"
    elif opponent.currentAction == "D":
        if random.random() <= p_r:
            self.currentAction = "D"
        else:
            self.currentAction = "C"
    elif opponent.currentAction == "C":
        self.currentAction = "C"
#         print("         " + self.name + ": " + self.currentAction)

class ProberRetaliator:
    def __init__(self, name):
        self.name = name
        self.payoff = 0
        self.currentAction = "C"
        self.injured = False
        self.attacked = False
        self.probe = False

    def move(self, opponent, u):
        if self.injured or u == 0:
            self.currentAction = "R"

        elif not self.probe: # if it's not currently probing
            if opponent.currentAction == "D":
                if random.random() <= p_r:
                    self.currentAction = "D"          # retaliates

            elif opponent.currentAction == "C":
                if random.random() <= p_p:
                    self.currentAction = "D"          # probes
                    self.probe = True
                else:
                    self.currentAction = "C"

        else: # if it is currently probing
            if opponent.currentAction == "D": # opponent has retaliated
                self.currentAction = "C" # reverts back to C
                self.probe = False
            elif opponent.currentAction == "C":
                self.currentAction = "D" # takes advantage and keeps probing

#         print(self.name + ": " + self.currentAction)

    def check(contestant, opponent):
        if contestant.currentAction == "D":
            if not isinstance(opponent, Mouse):
                if random.random() <= p_i: # 1 in 10 chance
                    opponent.payoff -= c # serious injury
                    opponent.injured = True
                else:
                    opponent.payoff -= s # scratch
            else:
                opponent.attacked = True

```

```

    if contestant.currentAction == "R":
        opponent.payoff += v
        return False

    return True

def main():

    """
    Hawk vs. Hawk
    """
    total = 0
    total2= 0

    for _ in range(num_trials):
        p1 = Hawk("p1")
        p2 = Hawk("p2")

        u = u_init #saved-time payoff starts at u_init

        while True:
            # p1 goes first
            p1.move()

            # check payoffs
            if not check(p1, p2):
                break

            # p2 goes next
            p2.move()

#            u -= u_minus

            # check payoffs
            if not check(p2, p1):
                break

            # decrease time-saved payoff
            u -= u_minus

        # uninjured players get time-saved payoff
        if u < 0:
            u = 0
        if not p1.injured:
            p1.payoff += u
        if not p2.injured:
            p2.payoff += u

        # add final payoffs to totals
        total += p1.payoff
        total2 += p2.payoff

    # calculate average of all trials, average of p1 and p2
    print("Hawk vs. Hawk avg payoff: " + str((total+total2)/(2*num_trials)))
    print("\n")
    hh = (total+total2)/(2*num_trials)

```

```

"""
Hawk vs. Mouse
"""
total = 0
total2 = 0
for _ in range(num_trials):
    p1 = Hawk("p1")
    p2 = Mouse("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move()

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move(u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Hawk vs. Mouse avg payoff: " + str(total/num_trials))
print("    Mouse's payoff: " + str(total2/num_trials))
print("\n")
hm1 = total/num_trials
mh2 = total2/num_trials

"""
Mouse vs. Hawk
"""

```

```

total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Mouse("p1")
    p2 = Hawk("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(u)

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move()

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Mouse vs. Hawk avg payoff: " + str(total/num_trials))
print("    Hawk's payoff: " + str(total2/num_trials))
print("\n")
mh1 = total/num_trials
hm2 = total2/num_trials

"""
Mouse vs. Mouse
"""
total = 0
total2 = 0

for _ in range(num_trials):

```

```

p1 = Mouse("p1")
p2 = Mouse("p2")

u = u_init

while True:
    # p1 moves
    p1.move(u)

    # check payoffs
    if not check(p1, p2):
        break

#         u -=u_minus

    # p2 moves
    p2.move(u)

    # check payoffs
    if not check(p2, p1):
        break

    # decrease time-saved payoff
    u -= u_minus

# uninjured players get time-saved payoff
if u < 0:
    u = 0
if not p1.injured:
    p1.payoff += u
if not p2.injured:
    p2.payoff += u

# add final payoffs to totals
total += p1.payoff
total2 += p2.payoff

# calculate average of num_trials trials
print("Mouse vs. Mouse avgavg payoff: " +
str((total+total2)/(2*num_trials)))
print("\n")
mm = (total+total2)/(2*num_trials)

"""
Retaliator vs. Mouse
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Retaliator("p1")
    p2 = Mouse("p2")

    u = u_init

```

```

while True:
    # p1 moves
    p1.move(p2, u)

    # check payoffs
    if not check(p1, p2):
        break

#         u -= u_minus

    # p2 moves
    p2.move(u)

    # check payoffs
    if not check(p2, p1):
        break

    # decrease time-saved payoff
    u -= u_minus

# uninjured players get time-saved payoff
if u < 0:
    u = 0
if not p1.injured:
    p1.payoff += u
if not p2.injured:
    p2.payoff += u

# add final payoffs to totals
total += p1.payoff
total2 += p2.payoff

# calculate average of num_trials trials
print("Retaliator vs. Mouse avg payoff: " + str(total/num_trials))
print("    Mouse's payoff: " + str(total2/num_trials))
print("\n")
rm1 = total/num_trials
mr2 = total2/num_trials

"""
Mouse vs. Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Mouse("p1")
    p2 = Retaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(u)

```

```

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Mouse vs. Retaliator avg payoff: " + str(total/num_trials))
print("    Retaliator's payoff: " + str(total2/num_trials))
print("\n")
mr1 = total/num_trials
rm2 = total2/num_trials

"""
Retaliator vs. Hawk
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Retaliator("p1")
    p2 = Hawk("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(p2, u)

        # check payoffs
        if not check(p1, p2):
            break

```



```

#             u -= u_minus

# p2 moves
p2.move()

# check payoffs
if not check(p2, p1):
    break

# decrease time-saved payoff
u -= u_minus

# uninjured players get time-saved payoff
if u < 0:
    u = 0
if not p1.injured:
    p1.payoff += u
if not p2.injured:
    p2.payoff += u

# add final payoffs to totals
total += p1.payoff
total2 += p2.payoff

# calculate average of num_trials trials
print("Retaliator vs. Hawk avg payoff: " + str(total/num_trials))
print("    Hawk's payoff: " + str(total2/num_trials))
print("\n")
rh1 = total/num_trials
hr2 = total2/num_trials

"""
Hawk vs. Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Hawk("p1")
    p2 = Retaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move()

        # check payoffs
        if not check(p1, p2):
            break

#             u -= u_minus

# p2 moves
p2.move(p1, u)

```

```

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Hawk vs. Retaliator avg payoff: " + str(total/num_trials))
print("    Retaliator's payoff: " + str(total2/num_trials))
print("\n")
hr1 = total/num_trials
rh2 = total2/num_trials

"""
Retaliator vs. Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Retaliator("p1")
    p2 = Retaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(p2, u)

        # check payoffs
        if not check(p1, p2):
            break

    #
        u -= u_minus

    # p2 moves
    p2.move(p1, u)

    # check payoffs
    if not check(p2, p1):
        break

```

```

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

    # calculate average of num_trials trials
    print("Retaliator vs. Retaliator avg payoff: " +
          str((total+total2)/(2*num_trials)))
    print("\n")
    rr = (total+total2)/(2*num_trials)

    """
    Prober-Retaliator vs. Mouse
    """
    total = 0
    total2 = 0

    for _ in range(num_trials):
        p1 = ProberRetaliator("p1")
        p2 = Mouse("p2")

        u = u_init

        while True:
            # p1 moves
            p1.move(p2, u)

            # check payoffs
            if not check(p1, p2):
                break

        #
            u -= u_minus

        # p2 moves
        p2.move(u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff

```

```

        if u < 0:
            u = 0
        if not p1.injured:
            p1.payoff += u
        if not p2.injured:
            p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Prober-Retaliator vs. Mouse avg payoff: " + str(total/num_trials))
print("    Mouse's payoff: " + str(total2/num_trials))
print("\n")
pml = total/num_trials
mp2 = total2/num_trials

"""
Mouse vs. Prober-Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Mouse("p1")
    p2 = ProberRetaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(u)

        # check payoffs
        if not check(p1, p2):
            break

    #
        u -= u_minus

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

# uninjured players get time-saved payoff
if u < 0:
    u = 0
if not p1.injured:
    p1.payoff += u

```

```

        if not p2.injured:
            p2.payoff += u

        # add final payoffs to totals
        total += p1.payoff
        total2 += p2.payoff

    # calculate average of num_trials trials
    print("Mouse vs. Prober-Retaliator avg payoff: " + str(total/num_trials))
    print("    Prober-Retaliator's payoff: " + str(total2/num_trials))
    print("\n")
    mp1 = total/num_trials
    pm2 = total2/num_trials

    """
    Prober-Retaliator vs. Hawk
    """
    total = 0
    total2 = 0

    for _ in range(num_trials):
        p1 = ProberRetaliator("p1")
        p2 = Hawk("p2")

        u = u_init

        while True:
            # p1 moves
            p1.move(p2, u)

            # check payoffs
            if not check(p1, p2):
                break

            #
            u -= u_minus

            # p2 moves
            p2.move()

            # check payoffs
            if not check(p2, p1):
                break

            # decrease time-saved payoff
            u -= u_minus

        # uninjured players get time-saved payoff
        if u < 0:
            u = 0
        if not p1.injured:
            p1.payoff += u
        if not p2.injured:
            p2.payoff += u

    # add final payoffs to totals

```

```

        total += p1.payoff
        total2 += p2.payoff

# calculate average of num_trials trials
print("Prober-Retaliator vs. Hawk avg payoff: " + str(total/num_trials))
print("    Hawk's payoff: " + str(total2/num_trials))
print("\n")
ph1 = total/num_trials
hp2 = total2/num_trials

"""
Hawk vs. Prober-Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Hawk("p1")
    p2 = ProberRetaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move()

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

# uninjured players get time-saved payoff
if u < 0:
    u = 0
if not p1.injured:
    p1.payoff += u
if not p2.injured:
    p2.payoff += u

# add final payoffs to totals
total += p1.payoff
total2 += p2.payoff

# calculate average of num_trials trials

```

```

print("Hawk vs. Prober-Retaliator avg payoff: " + str(total/num_trials))
print("    Prober-Retaliator's payoff: " + str(total2/num_trials))
print("\n")
hp1 = total/num_trials
ph2 = total2/num_trials

"""
Prober-Retaliator vs. Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = ProberRetaliator("p1")
    p2 = Retaliator("p2")

    u = u_init

    while True:
        p1.move(p2, u)

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals

    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Prober-Retaliator vs. Retaliator avg payoff: " +
str(total/num_trials))
print("    Retaliator's payoff: " + str(total2/num_trials))
print("\n")

```

```

pr1 = total/num_trials
rp2 = total2/num_trials

"""
Retaliator vs. Prober-Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = Retaliator("p1")
    p2 = ProberRetaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(p2, u)

        # check payoffs
        if not check(p1, p2):
            break

#         u -= u_minus

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("Retaliator vs. Prober-Retaliator avg payoff: " +
str(total/num_trials))
print("    Prober-Retaliator's payoff: " + str(total2/num_trials))
print("\n")
rp1 = total/num_trials
rp2 = total2/num_trials

```



```

"""
Prober-Retaliator vs. Prober-Retaliator
"""
total = 0
total2 = 0

for _ in range(num_trials):
    p1 = ProberRetaliator("p1")
    p2 = ProberRetaliator("p2")

    u = u_init

    while True:
        # p1 moves
        p1.move(p2, u)

        # check payoffs
        if not check(p1, p2):
            break

        # p2 moves
        p2.move(p1, u)

        # check payoffs
        if not check(p2, p1):
            break

        # decrease time-saved payoff
        u -= u_minus

    # uninjured players get time-saved payoff
    if u < 0:
        u = 0
    if not p1.injured:
        p1.payoff += u
    if not p2.injured:
        p2.payoff += u

    # add final payoffs to totals
    total += p1.payoff
    total2 += p2.payoff

# calculate average of num_trials trials
print("PR vs. PR avg payoff: " + str((total+total2)/(2*num_trials)))
print("\n")
pp = (total+total2)/(2*num_trials)

"""
Print out-puts as table
"""
table = [["", "Mouse", "Hawk", "Retaliator", "PR"],
          ["Mouse", mm, (mh1+mh2)/2, (mr1+mr2)/2, (mp1+mp2)/2],
          ["Hawk", (hm1+hm2)/2, hh, (hr1+hr2)/2, (hp1+hp2)/2],

```

```

        ["Retaliator", (rm1+rm2)/2, (rh1+rh2)/2, rr, (rp1+rp2)/2],
        ["PR", (pm1+pm2)/2, (ph1+ph2)/2, (pr1+pr2)/2, pp]]
print(tabulate(table, headers = "firstrow"))

if __name__ == "__main__":
    main()

```

Output used to generate Table 6 (num_trials = 100000):

```

Hawk vs. Hawk avg payoff: -20.1133125

Hawk vs. Mouse avg payoff: 80.0
    Mouse's payoff: 20.0

Mouse vs. Hawk avg payoff: 19.5
    Hawk's payoff: 79.5

Mouse vs. Mouse avgavg payoff: 30.0

Retaliator vs. Mouse avg payoff: 0.0
    Mouse's payoff: 60.0

Mouse vs. Retaliator avg payoff: 0.0
    Retaliator's payoff: 60.0

Retaliator vs. Hawk avg payoff: -25.050225
    Hawk's payoff: -15.5846

Hawk vs. Retaliator avg payoff: -14.90955
    Retaliator's payoff: -25.412825

Retaliator vs. Retaliator avg payoff: 30.0

Prober-Retaliator vs. Mouse avg payoff: 63.982175
    Mouse's payoff: 19.426175

Mouse vs. Prober-Retaliator avg payoff: 11.266175
    Prober-Retaliator's payoff: 71.266175

Prober-Retaliator vs. Hawk avg payoff: -26.092925
    Hawk's payoff: -14.32605

Hawk vs. Prober-Retaliator avg payoff: -15.49565
    Prober-Retaliator's payoff: -25.26095

Prober-Retaliator vs. Retaliator avg payoff: -6.2448
    Retaliator's payoff: 33.412425

Retaliator vs. Prober-Retaliator avg payoff: -9.1115

```

Prober-Retaliator's payoff: 35.982575

PR vs. PR avg payoff: 3.0617

Sample table output (with $p_i = 0.90$ and $\text{num_trials} = 100000$):

	Mouse	Hawk	Retaliator	Prober-Retaliator
-----	-----	-----	-----	-----
Mouse	30	19.75	30	15.3448
Hawk	79.75	-10.136	63.5529	59.6246
Retaliator	30	-84.0495	30	-69.4299
Prober-Retaliator	67.6996	-80.1159	53.8537	-11.6767