PART B - PROBLEM 4: IMPLEMENTING A SIMULATION WITH DRUGS (10/10 points)

In this problem, we consider the effects of both administering drugs to the patient and the ability of virus particle offsprings to inherit or mutate genetic traits that confer drug resistance. As the virus population reproduces, mutations will occur in the virus offspring, adding genetic diversity to the virus population. Some virus particles gain favorable mutations that confer resistance to drugs.

RESISTANTVIRUS CLASS

In order to model this effect, we introduce a subclass of `SimpleVirus` called `ResistantVirus`. `ResistantVirus` maintains the state of a virus particle's drug resistances, and accounts for the inheritance of drug resistance traits to offspring. Implement the `ResistantVirus` class.

```
#
# PROBLEM 4
#
class ResistantVirus(SimpleVirus):
    """
    Representation of a virus which can have drug resistance.
    """

    def __init__(self, maxBirthProb, clearProb, resistances, mutProb):
        """
        Initialize a ResistantVirus instance, saves all parameters as attributes
        of the instance.

        maxBirthProb: Maximum reproduction probability (a float between 0-1)

        clearProb: Maximum clearance probability (a float between 0-1)
```

Correct

Test results

Hide output

CORRECT

```
Test: ResistantVirus 0
```

Create a ResistantVirus that is never cleared and always reproduces.

Output:

```
virus = ResistantVirus(1.0, 0.0, {}, 0.0)
Testing update() and doesClear(); virus should not be cleared and should always reproduce.
Test completed.
```

```
Test: ResistantVirus 1
```

Create a ResistantVirus that is never cleared and never reproduces.

Output:

```
virus = ResistantVirus(0.0, 0.0, {}, 0.0)
Testing update() and doesClear(); virus should not be cleared and should never reproduce.
Test completed.
```

`Test: ResistantVirus 2`

Create a ResistantVirus that is always cleared and always reproduces.

Output:

```
virus = ResistantVirus(1.0, 1.0, {}, 0.0)
Testing update() and doesClear(); virus should always be cleared and should always reproduce.
Test completed.
```

`Test: ResistantVirus 3`

Create a ResistantVirus that is always cleared and never reproduces.

Output:

```
virus = ResistantVirus(0.0, 1.0, {}, 0.0)
Testing update() and doesClear(); virus should always be cleared and should never reproduce.
Test completed.
```

`Test: ResistantVirus 4`

Test for virus resistances.

Output:

```
virus = ResistantVirus(0.0, 1.0, {"drug1":True, "drug2":False}, 0.0)
Running virus.reproduce(0, []) to make sure that resistances are not changed.
Test completed.
```

`Test: ResistantVirus 5`

Test for virus reproduction with drugs applied.

Output:

```
virus = ResistantVirus(1.0, 0.0, {"drug1":True, "drug2":False}, 0.0)
Test: child = virus.reproduce(0, ["drug2"])
Test: child = virus.reproduce(0, ["drug1"])
Test completed.
```

`Test: ResistantVirus 6`

Check that mutProb is applied correctly in the reproduce step.

Output:

```
virus = ResistantVirus(1.0, 0.0, {'drug1':True, 'drug2': True, 'drug3': True, 'drug4': True,
'drug5': True, 'drug6': True}, 0.5)
Reproducing 10 times by calling virus.reproduce(0, [])
Checking the resistances of the children to each of the 6 prescriptions.
Since mutProb = 0.5 and the parent virus was resistant to all 6 drugs, we expect each child
to be resistant to, on average, half of the six drugs.
Test completed.
```

Help

```
Test: ResistantVirus 7
```

Test for positive mutability.

Output:

```
virus = ResistantVirus(1.0, 0.0, {"drug2": True}, 1.0)
Making 100 successive generations and testing their resistance to drug2
Test completed.
```

```
Test: ResistantVirus 8
```

Test for negative mutability.

Output:

```
virus = ResistantVirus(1.0, 0.0, {"drug1": True}, 0.0)
Making 100 successive generations and testing their resistance to drug1.
Test completed.
```

```
Test: ResistantVirus 9
```

Test for virus reproduction with drugs applied.

Output:

```
virus = ResistantVirus(0.0, 0.0, {"drug1":True, "drug2":False}, 0.0)
Test: child = virus.reproduce(0, ["drug2"])
Test: child = virus.reproduce(0, ["drug1"])
Test completed.
```

Hide output

Check Save *You have used 4 of 30 submissions*

Show Discussion New Post