MITx: 6.00.2x Introduction to Computational Thinking and Data Science

sandipan_dey ▾

Courseware | Updates & News | Calendar | Wiki | Discussion | Progress

Help

PART B - PROBLEM 4: TREATEDPATIENT CLASS (10/10 points)

We also need a representation for a patient that accounts for the use of drug treatments and manages a collection of `ResistantVirus` instances. For this, we introduce the `TreatedPatient` class, which is a subclass of `Patient`. `TreatedPatient` must make use of the new methods in `ResistantVirus` and maintain the list of drugs that are administered to the patient.

Drugs are given to the patient using the TreatedPatient class's `addPrescription()` method. What happens when a drug is introduced? The drugs we consider **do not directly kill virus particles lacking resistance to the drug**, but prevent those virus particles from reproducing (much like actual drugs used to treat HIV). Virus particles with resistance to the drug continue to reproduce normally. Implement the `TreatedPatient` class.

Hint: `reproduce` function child resistances

```
# Enter your definitions for the ResistantVirus and TreatedPatient classes in this box.
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).

        resistances: A dictionary of drug names (strings) mapping to the state
```

Correct

Test results

Hide output

CORRECT

`Test: TreatedPatient 1`

Create a TreatedPatient with virus that is never cleared and always reproduces.

Output:

```
virus = ResistantVirus(1.0, 0.0, {}, 0.0)
patient = TreatedPatient([virus], 100)
Updating patient for 100 time steps
Test completed.
```

`Test: TreatedPatient 2`

Create a TreatedPatient with virus that is always cleared and always reproduces.

Output:

```
virus = ResistantVirus(1.0, 1.0, {}, 0.0)
patient = TreatedPatient([virus], 100)
Updating patient for 100 time steps
Test completed.
```

`Test: TreatedPatient 3`

Test for adding duplicate prescriptions in TreatedPatient

Output:

```
Test completed.
```

`Test: TreatedPatient 4`

Test addPrescription and getPrescription in TreatedPatient.

Output:

```
patient = TreatedPatient([], 100)
Adding prescription Drug P
Drug P in plist: True
Adding prescription Drug N
Drug N in plist: True
Adding prescription Drug K
Drug K in plist: True
Adding prescription Drug O
Drug O in plist: True
Adding prescription Drug M
Drug M in plist: True
Adding prescription Drug Q
Drug Q in plist: True
Adding prescription Drug B
Drug B in plist: True
Adding prescription Drug J
Drug J in plist: True
Adding prescription Drug A
Drug A in plist: True
Adding prescription Drug I
Drug I in plist: True
Adding prescription Drug P
Drug P in plist: True
len(patient.getPrescriptions()): 10
Adding prescription Drug N
Drug N in plist: True
len(patient.getPrescriptions()): 10
Adding prescription Drug K
Drug K in plist: True
len(patient.getPrescriptions()): 10
Adding prescription Drug O
Drug O in plist: True
len(patient.getPrescriptions()): 10
Adding prescription Drug M
Drug M in plist: True
len(patient.getPrescriptions()): 10
Test completed.
```

`Test: TreatedPatient 5`

Test of getting TreatedPatient's resistant pop

Output:

```
virus1 = ResistantVirus(1.0, 0.0, {"drug1": True}, 0.0)
virus2 = ResistantVirus(1.0, 0.0, {"drug1": False, "drug2": True}, 0.0)
virus3 = ResistantVirus(1.0, 0.0, {"drug1": True, "drug2": True}, 0.0)
patient = sm.TreatedPatient([virus1, virus2, virus3], 100)
patient.getResistPop(['drug1']): 2
patient.getResistPop(['drug2']): 2
patient.getResistPop(['drug1','drug2']): 1
patient.getResistPop(['drug3']): 0
patient.getResistPop(['drug1', 'drug3']): 0
patient.getResistPop(['drug1','drug2', 'drug3']): 0
Test completed.
```

Test: TreatedPatient 6

Test for virus populations in TreatedPatient.

Output:

```
virus1 = ResistantVirus(1.0, 0.0, {"drug1": True}, 0.0)
virus2 = ResistantVirus(1.0, 0.0, {"drug1": False}, 0.0)
patient = TreatedPatient([virus1, virus2], 1000000)
patient.addPrescription("drug1")
Updating patient 5 times
Expect resistant population to be 2^5 +/- 10
Expect total population to be the resistant population plus 1
Test completed.
```

Hide output

Check Save *You have used 5 of 30 submissions*

Show Discussion

New Post

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+

Privacy Policy (Revised 4/16/2014)

Jobs at edX