REFRESHER ON GENERATORS

Here is the lecture from 6.00.1x on generators. Additionally, you can also take a look at Chapter 8.3 in the textbook.

For the following problem, consider the following way to write a power set generator. The number of possible combinations to put n items into one bag is 2^n. Here, items is a Python list. If need be, also check out the docs on bitwise operators (<<, >>, &, |, ~, ^).

```
# generate all combinations of N items
def powerSet(items):
    N = len(items)
    # enumerate the 2**N possible combinations
    for i in xrange(2**N):
        combo = []
        for j in xrange(N):
            # test bit jth of integer i
            if (i >> j) % 2 == 1:
                combo.append(items[j])
        yield combo
```

L8 PROBLEM 4 (10/10 points)

As above, suppose we have a generator that returns every combination of objects in one bag. We can represent this as a list of 1s and 0s denoting whether each item is in the bag or not.

Write a generator that returns every arrangement of items such that each is in one or none of two different bags. Each combination should be given as a tuple of two lists, the first being the items in bag1, and the second being the items in bag2.

```
def yieldAllCombos(items):
    """
      Generates all combinations of N items into two bags, whereby each
      item is in one or zero bags.

      Yields a tuple, (bag1, bag2), where each bag is represented as
      a list of which item(s) are in each bag.
    """
```

Note this generator should be pretty similar to the `powerSet` generator above.

We mentioned that the number of possible combinations for N items into one bag is 2^n. How many possible combinations exist when there are two bags? Think about this for a few minutes, then click the following hint to confirm if your guess is correct. Remember that a given item can only be in bag1, bag2, or neither bag -- it is not possible for an item to be present in both bags!

How many possible combinations exist for N items into two bags?

```
# generate all combinations of N items
def yieldAllCombos(items):
    N = len(items)
    # enumerate the 3**N possible combinations
    def base10toN(num,n):
        return ((num == 0) and  "0" ) or ( base10toN(num // n, n).lstrip("0") + "0123456789abcdefghijklmnopqrst
    for i in xrange(3**N):
        bag1, bag2 = [], []
        i3 = base10toN(i, 3).zfill(N)
```

Correct

Test results

Hide output

CORRECT

Test: 1

In this test we call buildItems() (from L18_code.py) then call your generator.

Output:

```
items = buildItems()
combos = yieldAllCombos(items)
Testing all the correct combinations exist in your solution
Test successfully completed.
```

Test: 2

In this test we call buildRandomItems() (from L18_code.py) then call your generator.

Output:

```
items = buildRandomItems(5)
combos = yieldAllCombos(items)
Testing all the correct combinations exist in your solution
Test successfully completed.
```

Hide output

Check | Show Answer

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

Privacy Policy (Revised 4/16/2014)

About edX

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+