MITx 6.00.2x
Introduction to Computational Thinking and Data Science

Help   shengtatng

Course   Progress   Dates   Discussion   Notes   Calendar

Course / UNIT 1 / Lecture 2 - Decision Trees and Dynamic Programming

Previous   Next

Exercise 1

Bookmark this page

Exercise 1

10.0/10.0 points (graded)

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 range(2**N):
        combo = []
        for j in range(N):
            # test bit jth of integer i
            if (i >> j) % 2 == 1:
                combo.append(items[j])
        yield combo
```

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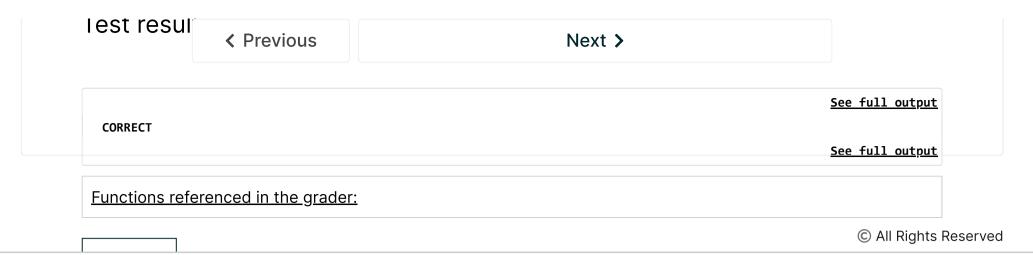
Note this generator should be pretty similar to the `powerSet` generator above.

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?

```
 1 def yieldAllCombos(items):
 2     """
 3
 4         Generates all combinations of N items into two bags, whereby each
 5         item is in one or zero bags.
 6
 7         Yields a tuple, (bag1, bag2), where each bag is represented as a list
 8         of which item(s) are in each bag.
 9     """
10     # Your code here
11     N = len(items)
12     # Enumerate the 3**N possible combinations
13     for i in range(3**N):
14         bag1 = []
15         bag2 = []
16         for j in range(N):
```

Press ESC then TAB or click outside of the code editor to exit

Correct

⊞ Calculator

Test resul

Previous Next >

See full output

CORRECT

See full output

Functions referenced in the grader:

edX

edX

About

Affiliates

edX for Business

Open edX

Careers

News

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

Sitemap

Connect

Blog

Contact Us

Help Center

Security

Media Kit

Calculator