BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 10 EXERCISE 10.8
ILLINOIS STATE UNIVERSITY

The (so-called) Birthday Paradox:
1. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
2. If there are 23 students in your class, what are the chances that two of you have the same birthday? You can estimate this probability by generating random samples of 23 birthdays and checking for matches. Hint: you can generate random birthdays with the randint function in the random module.
You can read about this problem at http: // en. wikipedia. org/ wiki/ Birthday_ paradox , and you can download my solution from http: // thinkpython. com/ code/ birthday. py .

#2

```
from __future__ import print_function, division
import random

def contains_duplicates(t):
    s = t[:]
    s.sort()
    for i in range(len(s)-1):
        if s[i] == s[i+1]:
            return True
    return False

def generate_random_birthdays(n):
    t = []
    for i in range(n):
        bday = random.randint(1, 365)
        t.append(bday)
    return t

def simulate_birthday_matches(num_students, num_simulations):
    count = 0
    for i in range(num_simulations):
        t = generate_random_birthdays(num_students)
        if contains_duplicates(t):
            count += 1
    return count

def main():
```

```
    num_students = 23
    num_simulations = 1000
    count = simulate_birthday_matches(num_students, num_simulations)

    print('After %d simulations' % num_simulations)
    print('with %d students' % num_students)
    print('there were %d simulations with at least one match' % count)

if __name__ == '__main__':
    main()


#1
print("*******1********")
def has_duplicates(lst):
    return len(lst) != len(set(lst))

print(has_duplicates([1, 2, 3, 4, 5]))
print(has_duplicates([1, 2, 3, 4, 5, 1]))
```

BIS420_PrajaktaPohare_Ch10_10.8.py

Users > prajaktapohare > Library > CloudStorage > OneDrive-ILStateUniversity > BIS420 > Week 10 > BIS420_PrajaktaPohare_Ch10_10.8.py > simulate_birthday_matches

```python
#2
from __future__ import print_function, division
import random

def contains_duplicates(t):
    s = t[:]
    s.sort()
    for i in range(len(s)-1):
        if s[i] == s[i+1]:
            return True
    return False

def generate_random_birthdays(n):
    t = []
    for i in range(n):
        bday = random.randint(1, 365)
        t.append(bday)
    return t

def simulate_birthday_matches(num_students, num_simulations):
    count = 0
    for i in range(num_simulations):
        t = generate_random_birthdays(num_students)
        if contains_duplicates(t):
            count += 1
    return count

def main():
    num_students = 23
    num_simulations = 1000
    count = simulate_birthday_matches(num_students, num_simulations)

    print('After %d simulations' % num_simulations)
    print('with %d students' % num_students)
    print('there were %d simulations with at least one match' % count)

if __name__ == '__main__':
    main()



#1
print("*******1*******")
def has_duplicates(lst):
    return len(lst) != len(set(lst))


print(has_duplicates([1, 2, 3, 4, 5]))
print(has_duplicates([1, 2, 3, 4, 5, 1]))
```