```
 1 import numpy as np
 2 from prettytable import PrettyTable
 3 import sys
 4
 5
 6
 7 def create_sequence_array(bound,
   num_suquences):
 8     """
 9     Creates a 50x10000 array of sequences
10     :param bound: int
11     :return: array of 10000 sequences
12     """
13
14     sequence_array = []
15
16     for i in range(num_suquences):
17         sequence_array.append([np.random.
   randint(0, bound) for x in range(50)])
18
19     return sequence_array
20
21
22 def find_calculated_average(bound,
   sequence_array, x):
23     """
24     :param bound: the bound
25     :param sequence_array: array of sequences
26     :param x: a number between 0 and bound
27     :return: calculated average
28     """
29
30     hits = 0
31
32     for sequence in sequence_array:
33
34         if x in sequence:
35             hits += 1
36
37
```

```
38      q = hits/10000
39
40      return 50 + (q - (q*50))/2
41
42
43  def find_real_average(bound, sequence_array,
    x):
44      """
45      :param bound: the bound
46      :param sequence_array: array of 10000
    sequences
47      :param x: a random number between 0 and
    bound
48      """
49      steps = 0
50
51      for sequence in sequence_array:
52
53          if x in sequence:
54              steps += sequence.index(x) + 1
55          else:
56              steps += 50
57
58      return steps / 10000
59
60
61  if __name__ == "__main__":
62      num_sequences = 10000
63      bound_array = (30, 50, 80, 100, 1000,
    10000, 100000, sys.maxsize)
64      Table = PrettyTable(["Bound", "Calculated
     Average", "Real Average"])
65
66      for bound in bound_array:
67          x = np.random.randint(0, bound)
68          sequence_array =
    create_sequence_array(bound, num_sequences)
69          calculated_average =
    find_calculated_average(bound, sequence_array
    , x)
```

```
70          real_average = find_real_average(
   bound, sequence_array, x)
71          if bound == sys.maxsize:
72              bound = "inf"
73          Table.add_row([bound,
   calculated_average, real_average])
74
75      print(Table)
76
```