

(a) Experimentally investigate how often one has to choose a random number between 1 and n until every number has appeared at least once. Conduct 100 experiments for $n = 10$, 100, and 1000, respectively, and determine the minimum, maximum, and average number of attempts for each n . An experiment consists of the necessary number of attempts until all numbers have been observed. (By the way, the expected value for the number of attempts is $n(1 + 1/2 + 1/3 + \dots + 1/n)$.)

```
import random

n = int(input("Enter the number:"))
def avr(n):
    attempt = 0
    numbers = list(range(1,n +1))
    def binary_search(numbers, random_number):
        n = len(numbers)

        def bin_search(left, right):
            if left >= right: # Stop point: element not found
                return None
            m_pos = (left + right) // 2
            m = numbers[m_pos]

            if random_number == m: # Found the number
                return m_pos
            elif random_number < m: # Search left half
                return bin_search(left, m_pos)
            else: # Search right half
                return bin_search(m_pos + 1, right)

        return bin_search(0, n)

    while numbers:
        random_number = random.randint(1, n)
        loc = binary_search(numbers, random_number) # Get the position of
the random number
        if loc is not None:
            del numbers[loc] # Remove that number from the list
            attempt += 1 # increase the attempt counter each time

    return attempt
avr_100_numbers = []
for i in range (1,100):
    attempt = avr(n) # Call the function
    avr_100_numbers.append(attempt)
    print(f"Attempt {i}: {attempt}")
avr_100 = sum(avr_100_numbers) / len(avr_100_numbers)
min_100 = min(avr_100_numbers)
max_100 = max(avr_100_numbers)
print ("Average:" ,avr_100, "\nMinimun:", min_100, "\nMaximum:", max_100)
# Output the average of attempts
```

