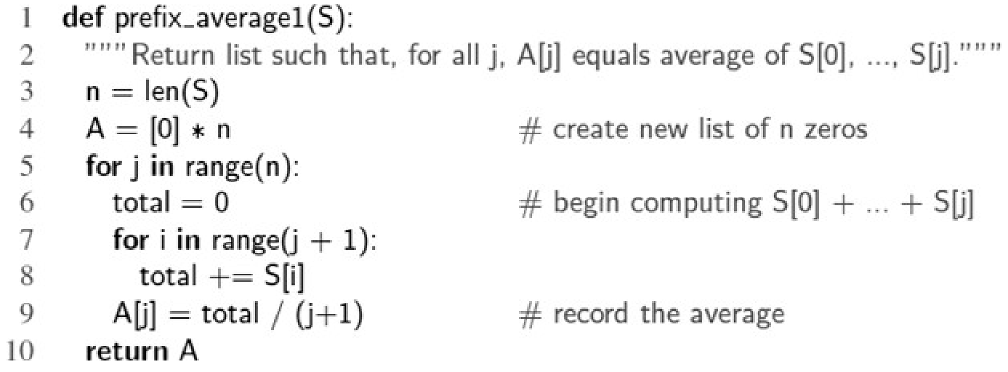
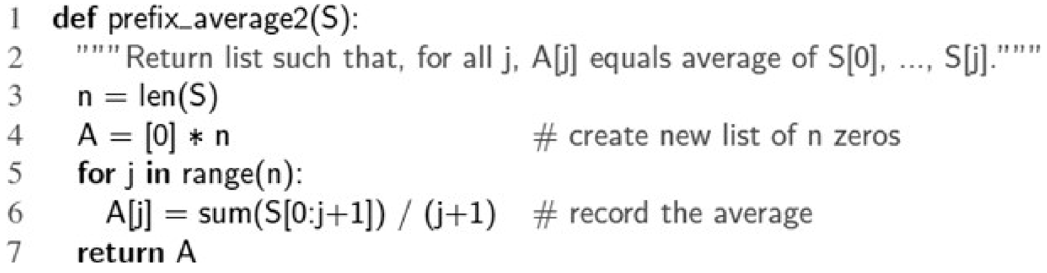
Problem 3. Experimental Analysis of Algorithms

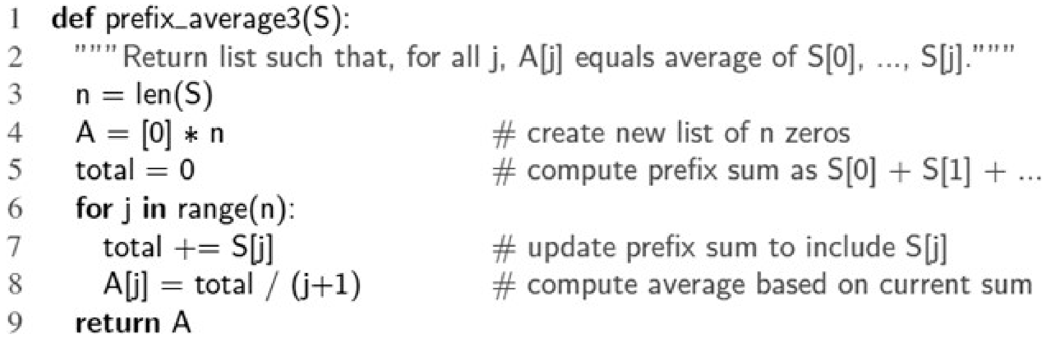
Perform an experimental analysis of the three algorithms: *prefix_average1*, *prefix_average2*, and *prefix_average3*. Visualize their running times as a function of input size with a log-log scale chart.



```
def prefix_average1(S):
  """Return list such that, for all j, A[j] equals average of S[0], ..., S[j]."""
  n = len(S)
  A = [0] * n                     # create new list of n zeros
  for j in range(n):
    total = 0                     # begin computing S[0] + ... + S[j]
    for i in range(j + 1):
      total += S[i]
    A[j] = total / (j+1)          # record the average
  return A
```



```
def prefix_average2(S):
  """Return list such that, for all j, A[j] equals average of S[0], ..., S[j]."""
  n = len(S)
  A = [0] * n                     # create new list of n zeros
  for j in range(n):
    A[j] = sum(S[0:j+1]) / (j+1)  # record the average
  return A
```



```
def prefix_average3(S):
  """Return list such that, for all j, A[j] equals average of S[0], ..., S[j]."""
  n = len(S)
  A = [0] * n                     # create new list of n zeros
  total = 0                       # compute prefix sum as S[0] + S[1] + ...
  for j in range(n):
    total += S[j]                 # update prefix sum to include S[j]
    A[j] = total / (j+1)          # compute average based on current sum
  return A
```