Algorithm Analysis Assignment

1. (5 pts) Give the big-O time complexity for the running time of each of the functions shown below (textbook exercises R-3.23 through R-3.27).


```
1  def example1(S):
2    """Return the sum of the elements in sequence S."""
3    n = len(S)
4    total = 0
5    for j in range(n):                # loop from 0 to n-1
6      total += S[j]
7    return total
8
9  def example2(S):
10   """Return the sum of the elements with even index in sequence S."""
11   n = len(S)
12   total = 0
13   for j in range(0, n, 2):          # note the increment of 2
14     total += S[j]
15   return total
16
17 def example3(S):
18   """Return the sum of the prefix sums of sequence S."""
19   n = len(S)
20   total = 0
21   for j in range(n):                # loop from 0 to n-1
22     for k in range(1+j):            # loop from 0 to j
23       total += S[k]
24   return total
25
26 def example4(S):
27   """Return the sum of the prefix sums of sequence S."""
28   n = len(S)
29   prefix = 0
30   total = 0
31   for j in range(n):
32     prefix += S[j]
33     total += prefix
34   return total
35
36 def example5(A, B):                 # assume that A and B have equal length
37   """Return the number of elements in B equal to the sum of prefix sums in A."""
38   n = len(A)
39   count = 0
40   for i in range(n):                # loop from 0 to n-1
41     total = 0
42     for j in range(n):              # loop from 0 to n-1
43       for k in range(1+j):          # loop from 0 to j
44         total += A[k]
45     if B[i] == total:
46       count += 1
47   return count
```

Code Fragment 3.10: Some sample algorithms for analysis.