3. a.

Initialise String result = 1; Give result value of “” = 1; Initialise “i” = 1; Give I value of 0 = 1; do n number of comparisons between i and n = n; increment i n number of times = n; do result + s n number of times = Xn; assign new result to result n number of times = n; return result once = 1;

T(n) = 1 + 1 + 1 + 1 + n + n + Xn + n + 1
T(n) = 3n + Xn + 5

b.

```
T(1) = 2445200.0 seconds
T(100) = 157200.0 seconds
T(1000) = 2903500.0 seconds (x10 -> x18) (1.8)
T(10000) = 7.05111E7 seconds (x10 -> x24) (2.4)
T(20000) = 1.815683E8 seconds (x2 -> x2.575) (1.29)
```

Ignoring the result from the first test, it is clear that the system is operating on Ө(n) because the ratios between n and the time are roughly 2 every single time, with a slightly inaccurate t(20000) test.

c.

```
T(1) = 71.707 seconds
T(100) = 1375.487 seconds (x100 -> 19.182) (0.2)
T(1000) = 13630.468 seconds (x10 -> 9.910) (0.99)
T(10000) = 110909.568 seconds (x10 -> 8.14) (0.81)
```

Ө(n) again because the ratio of the time to n seems to be roughly 1 every time: even though the program uses a nested loop, dividing by the number of repeats evens this out and you still get the same complexity.