DS&A PS1 Q3

Part A

- Assignment of “result” – 1 elementary operation
- Assignment of “I” – 1 elementary operation
- Comparison of “I” to “n” – n + 1 operations
- Increment “I” n times – n operations
- Perform concatenation operation – Xn operations
- Assignment of concatenated result – n operations
- Return of result – 1 elementary operation

Therefore:

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

T(n) = Xn + 3n + 4

Part B

The complexity class for the unoptimized repeatString() method is Big-Theta **$\Theta(n^2)$**. This is because Strings in Java are **immutable**. This is to say that the state of the object cannot be changed after its initialisation.

Part C

The complexity class for the optimized repeatString() method is Big-Theta **$\Theta(n)$**. This is due to the StringBuffers ability to append a new string to itself in $\Theta(1)$ time. Repeated n times, we end up with complexity of $\Theta(n)$.