Question 3a:

```
public class StringRepeater {
    public String repeatString(String s, int n){
        String result = "";
        for(int i=0; i<n; i++) {
            result = result + s;
        }
        return result;
    }
}
```

String result = ""	1
int i=0	1
i<n	N+1
i++	N
result + s;	(X)*N *My guess ~ (5*N)*N - Word character Length "Hello" Multiplied by iterations.*
Result=…	N
return result;	1
Total	$T(N) = XN + 3N + 4$

Question 3B:

$$\emptyset(N^2)$$

This function has a quadratic Big Theta complexity because each time the string is concatenated it splits the string up linearly into characters and the characters of the string grow linearly with each iteration, which makes a quadratic sequence – this is the highest order of complexity for the algorithm.

Question 3C:

$$\emptyset(N)$$

This function Has a linear Big Theta complexity because the StringBuffer is mutable so it takes a constant amount of cycles/time to change the variable/concatenate it – this means that the highest order is the "for loop" that linearly appends the same word onto itself.