Juan Marin - JM2797
CM 20254Data Structures and Algorithms
Problem sheet 1 question 2

2a) unoptimized repeatString():

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

Computing T(n):
Creating string result - +1
Initialize int i once - +1
Perform n+1 comparisons of i between 0 and n - +(n+1)
Increment the variable *i,* n times - +n
Concatenating String s to result n times + X(n) (function which will add complexity depending on size of s) – +(n +X(n))
Returning result - +1

Total T(n): 3n + 4 + X(n)

2b) To find the complexity class of my unoptimized repeatString() I recorded the time taken for n=1, 100, 500, 1000, 1500, 2000, 2500, 5000 and 10000, repeating each n and finding the average, the standard deviation and using an confidence interval of 95%, with 20,000 repetitions for each n. These are the results:

| n | Time taken (average)/seconds | T(n)/T(1) |
|---|---|---|
| 1 | 0.0000013628769 | 1 |
| 100 | 0.00000854110445 | 6.27 |
| 500 | 0.00000600781065 | 72.4 |
| 1000 | 0.00024960970365 | 183 |
| 1500 | 0.0005709921591 | 429 |
| 2000 | 0.0008894715218 | 653 |
| 2500 | 0.0013733629162 | 1001 |
| 5000 | 0.0065211313841 | 4780 |
| 10000 | 0.0222380713754 | 16300 |

With this data I can come to the conclusion that the complexity class is $\Theta(n^2)$. I came to this conclusion by examining the relation between the average time taken for n=1 and bigger results like n=2500 n=5000 and n=10000. By using the formula $T=an^2$ for some number a we can see that when we carry out the operation $T(n)/T(1)/n^2$ we get the following results: 0.0001606, 0.0001683 and 0.00163, meaning that the multiple of $n^{2,}$ a is close to that and the rest of the formula (bn + c) can be ignored.

2c) To find the complexity class of my optimized repeatString() I recorded the time taken for n=1, 100, 1000, 10000, 50000, 100000 repeating each n and finding the average with 50,000 repetitions for each n. These are the results:

| n | Time taken (average)/seconds | T(n)/T(1) |
|---|---|---|
| 1 | 5.95E-08 | 1 |
| 100 | 1.55E-06 | 26 |
| 1000 | 9.20E-06 | 155 |
| 10000 | 8.38E-05 | 1410 |
| 50000 | 3.93E-04 | 6610 |
| 100000 | 1.09E-03 | 18400 |

With this data I can come to the conclusion that the complexity class is Θ(n). I came to this conclusion by examining the relation between the average time taken for n=1 and bigger results like n=10000 n=50000 and n=100000. By using the formula T=an for some number a we can see that when we carry out the operation T(n)/T(1)/n we get the following results: 0.141, 0.132 and 0.184, meaning that the multiple of n a is close to that and the rest of the formula (+ c) can be ignored.