COS126 Repeat.java (§4.1 Performance)

```java
/*************************************************************************
 *  Compilation:  javac-introcs Repeat.java
 *  Execution:    java-introcs Repeat N
 *
 *  Create a string that contains N copies of the letter x.
 *
 *************************************************************************/
public class Repeat {
    public static String method1(int N) {
        if      (N == 0) return "";
        String temp = method1(N / 2);
        if (N % 2 == 0) return temp + temp;
        else            return temp + temp + "x";
    }
    public static String method2(int N) {
        String s = "";
        for (int i = 0; i < N; i++)
            s = s + "x";
        return s;
    }
    public static String method3(int N) {
        if      (N == 0) return "";
        else if (N == 1) return "x";
        else return method3(N / 2) + method3(N - (N / 2));
    }
    public static String method4(int N) {
        char[] temp = new char[N];
        for (int i = 0; i < N; i++)
            temp[i] = 'x';
        return new String(temp);
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);

        Stopwatch timer = new Stopwatch();
        String s = method4(N);
        System.out.println("Elapsed time for method 4 = " + timer.elapsedTime());

        timer = new Stopwatch();
        s = method3(N);
        System.out.println("Elapsed time for method 3 = " + timer.elapsedTime());

        timer = new Stopwatch();
        s = method1(N);
        System.out.println("Elapsed time for method 1 = " + timer.elapsedTime());

        timer = new Stopwatch();
        s = method2(N);
        System.out.println("Elapsed time for method 2 = " + timer.elapsedTime());
    }

}
```