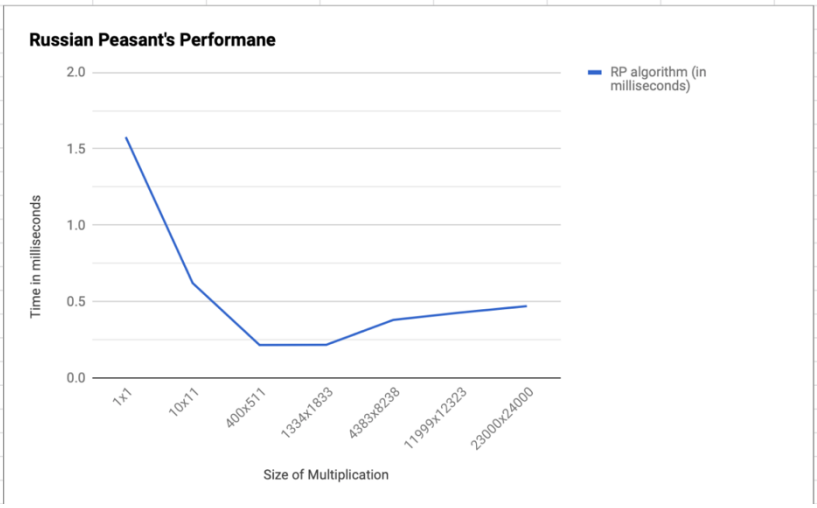
The code I made to implement the Russian Peasants algorithm in java.

```java
public static int algorithm(int n, int m){

    int accumulator = 0;

    while(n>1){
        if(n%2 == 0){
            n = n/2;
            m = m*2;
        }
        else {
            n = (n - 1) / 2;
            accumulator = accumulator + m;
            m = m * 2;
        }

    }

    return accumulator + m;
}
```

N	RP algorithm (in milliseconds)
1x1	1.58
10x11	0.62
400x511	0.22
1334x1833	0.22
4383x8238	0.38
11999x12323	0.43
23000x24000	0.47



As seen above, I created a graph which displays the results of my java implementation of the “Russian Peasant’s algorithm”. We can see that smaller integers actually take longer to compute a result from. This is because the given integers are so small that the computer will have to work very little to compute a result and as such it decides to prioritise other running processes over the algorithm. However when the integers start growing in size, we can see the amount of elapsed time lower as the OS starts to prioritise the algorithm more. The elapsed time starts to increase again as the size of the integers gets bigger.