HW2

1.7) $O(nm)$ or $n \cdot m$ time.

We have to go through each bit of whichever acts as y, because the division by 2 only knocks off one bit each time. Let's say that's m. Then for each call, we have to perform the division, the check for odd/even, the multiplication, & possibly the addition, if y is odd. This is then-bit.

1.25) $2^{125} \pmod{127}$

$2 \cdot 2^{125} \pmod{127} = 1 \pmod{127}$ Fermat's little theorem

$\frac{2 \cdot 2^{125} \pmod{127}}{2} = \frac{128 \pmod{127}}{2}$

$2^{125} \pmod{127} = 64 \pmod{127}$

$2^{21} \pmod{18}$

$2^{21} \pmod{18} = 8$

X	Y	Yodd	Z	Return value
2	21	1	16	$512 \pmod{18} = 8$
2	10	0	14	$196 \pmod{18} = 16$
2	5	1	4	$32 \pmod{18} = 14$
2	2	0	2	4
2	1	1	1	2
2	0	0	N/A	1

X	y	yodd	Z	return
2	125	1	64	$64^2 \bmod(127) = 64$ → same equivalence class
2	62	0	8	64
2	31	1	2	8
2	15	1	1	2
2	7	1	8	$128 \pmod{127} = 1$
2	3	1	2	$8 \pmod{127}$
2	1	1	1	$2 \pmod{127}$
2	0	0	NA	$1 \pmod{127}$