Lab Week 4

Questions regarding material:

Homework 2
Problem 2: Assume that we are using military time: 00:00 -> 23:59

A. What time would it be 300 hours after 1:35

   0 → 23 hours. This makes a total of 24 possible values.
   0 → 59 minutes. This makes a total of 60 possible values.

   300 hours mod 24 = 12 hours
   1:35 + 12 = 13:35

B. What time would it be 150,000 minutes after 23:30

   150,000/60 = 2500 hours in the 150,000 minutes

   150,000 mod 60 = 0 → no minutes to add

   2500 mod 24 = 4. So, we add hours: 23:30 + 4 = 3:30

Homework 3
Problem 7: Find the last digit of 7^(2021) mod 10 knowing that $7^2 \equiv 49 \equiv -1 \pmod{10}$

Assumig we are modding 144 mod 10
(12)^2 mod 10 → 2^2 = 4. Where 12 mod 10 = 2
12 * 12 mod 10 is the same as 2 * 12 mod 10 which is the same 2 * 2 mod 10
12 mod 10 = 2

Use this same approach for 7^(2021) mod 10

Hashing Functions:

Very USEFUL → O(1) access. And insertion costs at most O(n).

O(1) = constant time. So immediate access

O(n) = linear time. Takes at most time proportional to your input size

Say we have this array:

0	1	2	3	4	5	6	7	8	9
128392	3847651	238189	22219	177649	38500	88	3	3676	2

Say your program asks you to grab 88:
Starting from the left: 7 tries
Starting from the right: 4 tries

If you created a hashtable and used mod 10, then it would take 1 TRY to get 88.
The Hashtable would look this:
Grabbing the numbers from left to right and modding by 10 to get their hash index
128392 mod 10 = 2
3847651 mod 10 = 1
238189 mod 10 = 9
..
88 mod 10 = 8

0	1	2	3	4	5	6	7	8	9
	3847651	128392						88	238189

As soon as we're to grab 88 from the hashtable, we simply do 88 mod 10 and we know exactly where 88 might be = index 8

Hashtables:

H(k) – our mapping function and K would be the return value, or key

0	1	2	3	4	5	6	7	8	9
22219	3847651	128392	177649					88	238189

What happens in a collision? Two or more numbers with the same h(k) value (key)

Linear Probing: When your index variable is already taken by another number. We simply move over to the next free slot.

Say we wish to add 22219 to our array. We see that index 9 is already taken by 238189. So, we move over to index 0. That is the next free slot.

Issues: We may run out of space! Not everything will fit!
We can have a bunch of data in random indexes.

If now we wish to insert 177649 = 9 when modded by 10.

We would insert in index 3 because that’s the next available index.