

# ECE345 A2

Jason Wang, Kevin Grafstrum, Oliver

February 10, 2017

## Question 7

$$\begin{aligned}j &= (j + i) \mod m & i = 0 \\&= ((j + i) \mod m + i) \mod m & i = 1 \\&= (((j + i) \mod m + i) \mod m + i) \mod m & i = 2 \\&= ((((j + i) \mod m + i) \mod m + i) \mod m + i) \mod m & i = 3 \\&= (j + i + (i - 1) + (i - 2) + (i - 3) + \cdots + (i - i)) \mod m \\&= (j + \sum_0^i n) \mod m \\&= (j + \frac{n^2}{2}) \mod m\end{aligned}$$

Thus  $c_1 = 1/2$  and  $c_2 = 0$  for  $h(k, i) = h'(k) + c_1 i + c_2 i^2$

## Question 8

**What is the size of your hash table and which hash function are you using?**

I am using a hash table of size 1000 and using unsigned long hash = 5381;  
For the hash function I used the **djb2** hashing algorithm, which seems like a form of double hashing.

```
unsigned long hash = 5381;
int tmp;

for(int i = 0; i < s.length() - 2; i++) {
    hash = ((hash << 5) + hash) + s[i];
}
return hash % list_size;
```

### Are you applying chaining or open addressing?

I am applying hashing by chaining. I used chaining so that the passwords that are close to each other will be on the same chain. This will evenly distribute all the possible passwords since they are random.

**Graph of load factor vs Collision Count (Experiments)**

lf	<i>Test1</i>	<i>Test2</i>	<i>Test3</i>	<i>Test4</i>	<i>Test5</i>	<i>Average</i>
1	171	174	172	149	170	167
0.5	86	93	93	75	91	87
0.33	56	62	71	58	72	63
0.25	49	42	54	45	45	47
0.2	42	55	52	35	41	45
0.17	24	34	44	25	37	32
0.14	30	27	22	12	31	24
0.13	29	23	27	26	21	25
0.11	23	18	30	21	27	23
0.1	20	30	29	13	23	23
$9 \cdot 10^{-2}$	21	20	22	20	20	20
$8.3 \cdot 10^{-2}$	15	14	22	16	21	17
$7.6 \cdot 10^{-2}$	16	23	23	12	19	18
$7.1 \cdot 10^{-2}$	16	16	11	9	8	12
$6.6 \cdot 10^{-2}$	10	18	20	14	19	16

