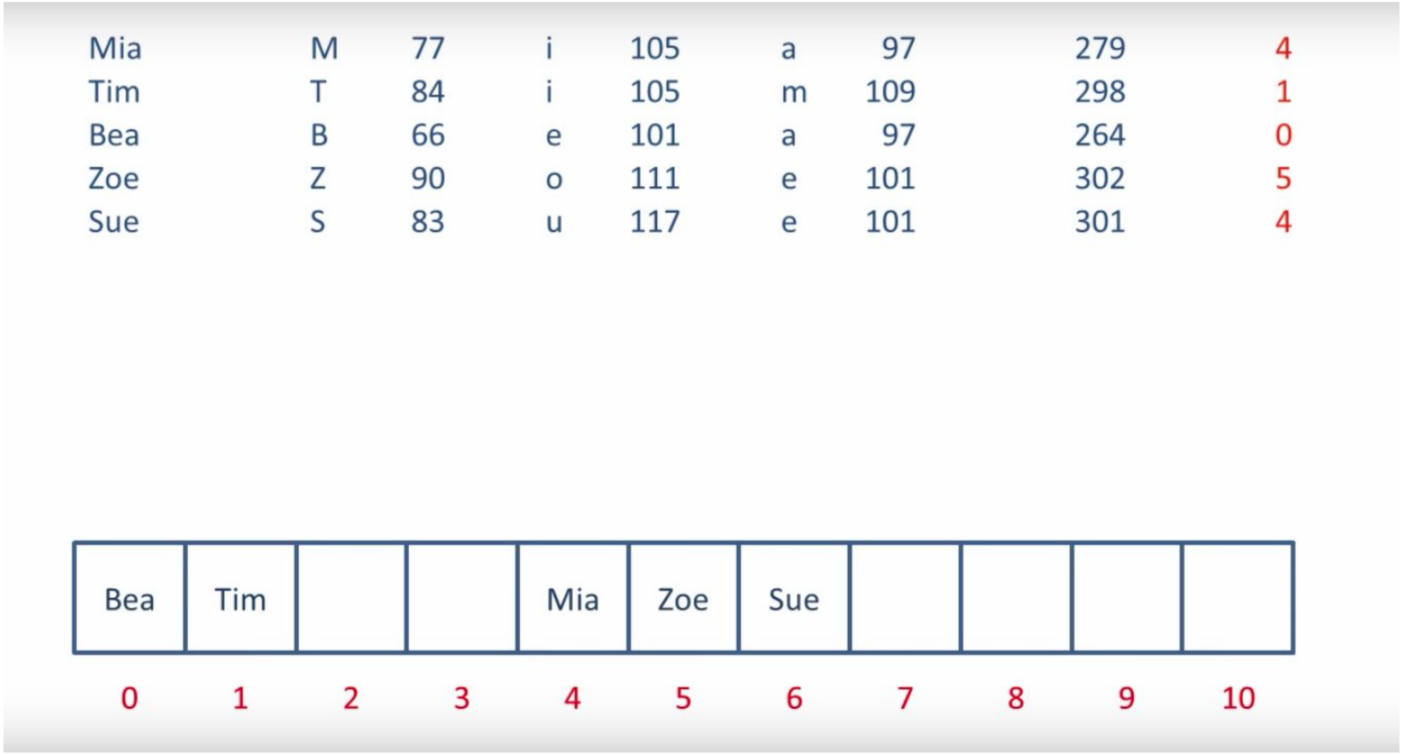
Developing our own Python Dictionary

Arnamoy Bhattacharyya

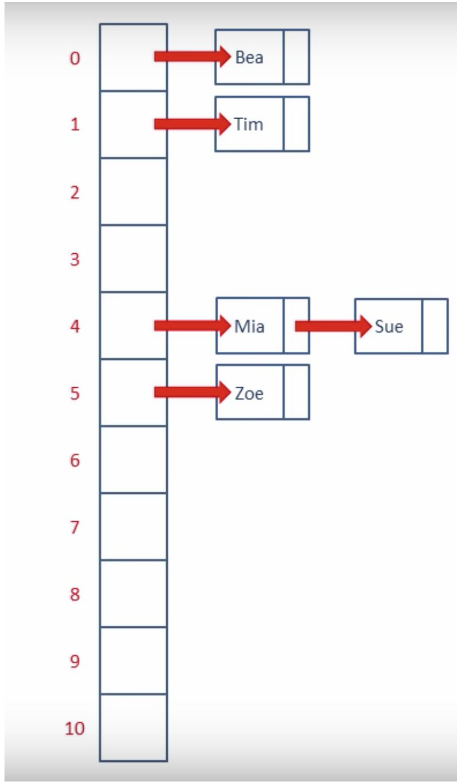
Collisions: Linear Probing



Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1
Bea	B	66	e	101	a	97	264	0
Zoe	Z	90	o	111	e	101	302	5
Sue	S	83	u	117	e	101	301	4

Bea	Tim			Mia	Zoe	Sue				
0	1	2	3	4	5	6	7	8	9	10

Collisions: Chaining



Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1
Bea	B	66	e	101	a	97	264	0
Zoe	Z	90	o	111	e	101	302	5
Sue	S	83	u	117	e	101	301	4

Collisions

More data → more probability for collision

Make the hash table big enough as compared to the number of keys

$$\text{Load Factor} = \frac{\text{Total number of items stored}}{\text{Size of the array}}$$

Resize the array based on certain threshold of *Load Factor (0.7 default for python)*

How doubling the size helps

- Collision contribution (chaining):
    - Two strings having the same “hashed” value
        - e.g. sue and use (both has sum of ASCII 333)
    - The size of the array generates the same modulo result
        - e.g. Tim and Len (for array length 11, both has modulo result 1)

Objectives of Hash Function

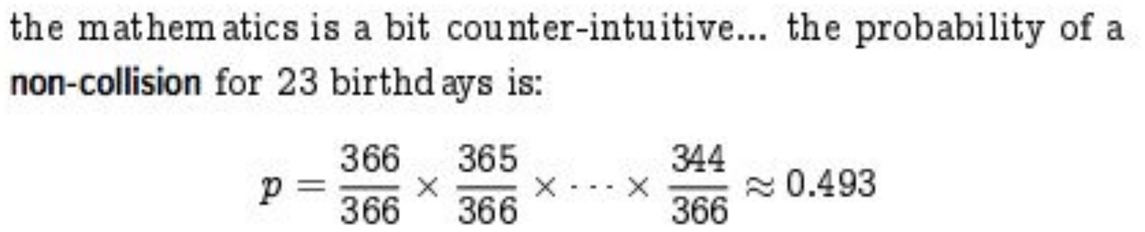
- Minimize collisions
- Uniform distribution of hash values
- Easy to calculate

How probable are collisions?

Highly Probable!

Birthday paradox:

Given a room of ‘n’ people, at least how many people you require to find among them any two people with matching birthdays with 50% probability?

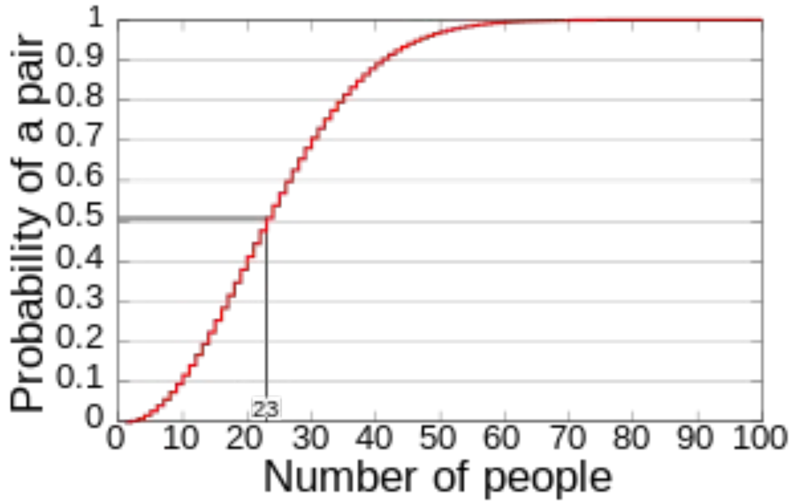


the mathematics is a bit counter-intuitive... the probability of a **non-collision** for 23 birthdays is:

$$p = \frac{366}{366} \times \frac{365}{366} \times \cdots \times \frac{344}{366} \approx 0.493$$

How probable are collisions?

Birthday Paradox



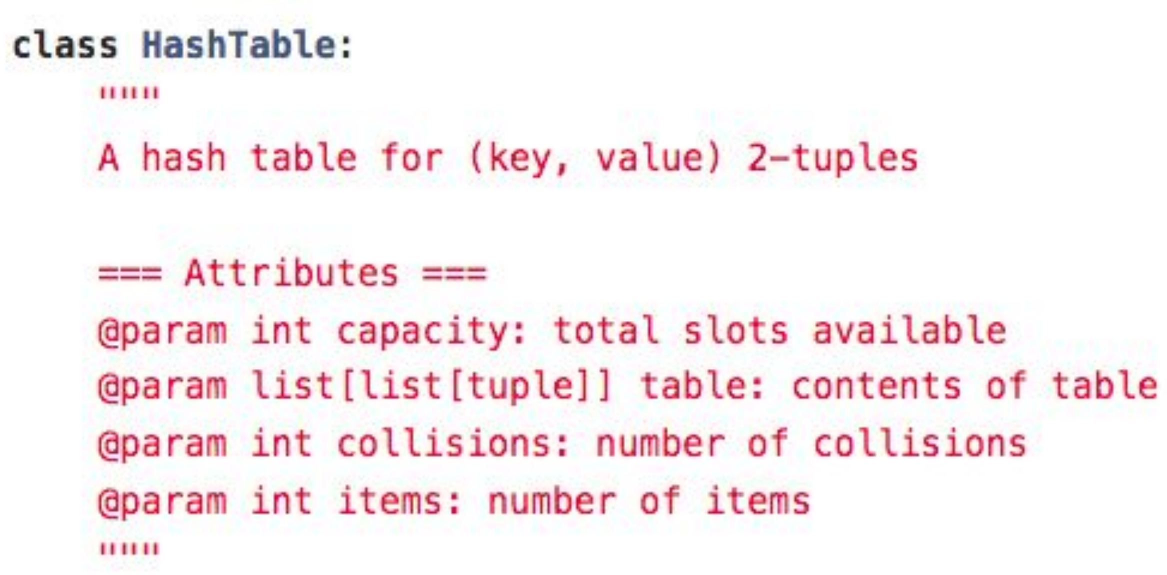
For 32-bit hash values: only 77,000 elements are required for significant risk (50%) of collision

Birthday Paradox

Onto pycharm

Coding our own hashtable

HashTable class



```
class HashTable:
    """
    A hash table for (key, value) 2-tuples

    === Attributes ===
    @param int capacity: total slots available
    @param list[list[tuple]] table: contents of table
    @param int collisions: number of collisions
    @param int items: number of items
    """
```

Coding HashTable

Onto PyCharm