

Hashing*

Arnamoy Bhattacharyya

*Some slides are taken from Danny Heap and Bogdan Simion, also youtube channel Kevin Drumm

Agenda

- Talk about hashing
- Course Evaluation

Lists and linear search

Data Structure	search()	insert()	delete()
List	$O(n)$	$O(n)$	$O(n)$

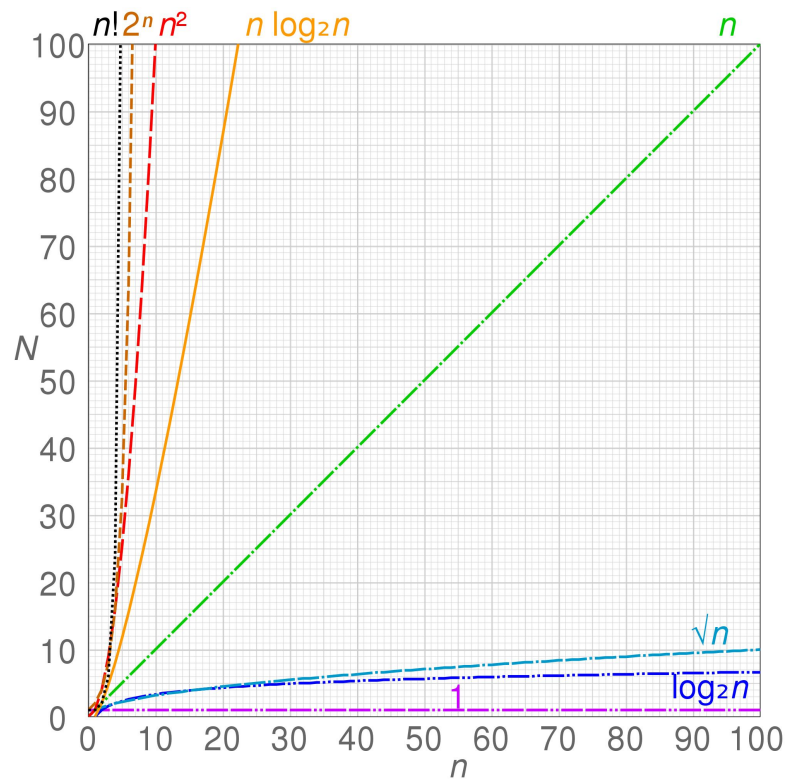
Ordered Lists and Binary search

Data Structure	search()	insert()	delete()
List	$O(n)$	$O(n)$	$O(n)$
Ordered List	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$

Demo

Onto Pycharm

Complexity Comparisons



How dictionary offers Constant time search?

By a technique called hashing

Example: hashing

What's the time complexity of:

- 1) Finding the string 'Ada' in this list?
- 2) What if you knew the index of 'Ada'?

Jan	Tim	Mia	Sam	Leo	Ted	Bea	Lou	Ada	Max	Zoe
0	1	2	3	4	5	6	7	8	9	10

Example: hashing

What's the time complexity of:

- 1) Finding the string 'Ada' in this list?
- 2) What if you knew the index of 'Ada'?
 - a) But how do you know the index of 'Ada' in the list?

Jan	Tim	Mia	Sam	Leo	Ted	Bea	Lou	Ada	Max	Zoe
0	1	2	3	4	5	6	7	8	9	10

Mia M i a

0 1 2 3 4 5 6 7 8 9 10

Populating the List

Mia

M

77

i

105

a

97

[illegible]

0

1

2

3

4

5

6

7

8

9

10

Populating the List

Mia	M	77	i	105	a	97	279			
0	1	2	3	4	5	6	7	8	9	10

Populating the List

Mia	M	77	i	105	a	97	279	4		
0	1	2	3	4	5	6	7	8	9	10

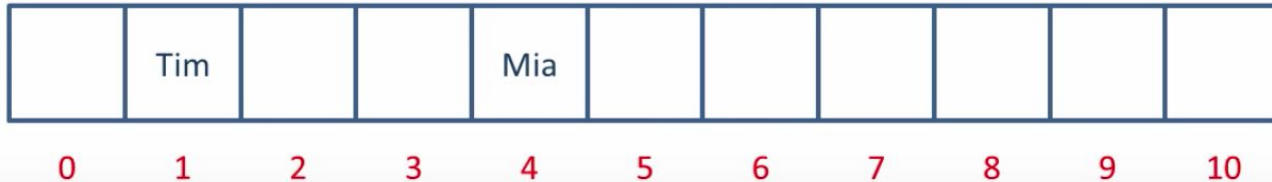
Populating the List

Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1

				Mia						
0	1	2	3	4	5	6	7	8	9	10

Populating the List

Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1



Populating the List

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

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

Populating the List

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
Jan	J	74	a	97	n	110	281	6
Ada	A	65	d	100	a	97	262	9
Leo	L	76	e	101	o	111	288	2
Sam	S	83	a	97	m	109	289	3
Lou	L	76	o	111	u	117	304	7
Max	M	77	a	97	x	120	294	8
Ted	T	84	e	101	d	100	285	10

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

Populating the List

Index number = $\text{sum ASCII codes} \text{ Mod } \text{size of array}$

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

Search

Find Ada $262 \text{ Mod } 11 = 9$

myData = Array(9)

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

Key, value pairs

Ada
10/12/1815

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

Key Value pairs

Ada
10/12/1815
English
Mathematician

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

Key:Value Pairs

Keys

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
27/01/1941	08/06/1955	31/12/1945	27/04/1791	20/02/1986	19/06/1978	13/02/1956	27/12/1922	23/04/1858	18/12/1815	17/06/1937
English	English	American	American	Russian	American	Polish	French	German	English	American
Astronomer	Inventor	Mathematician	Inventor	Space Station	Actress	Logician	Biologist	Physicist	Mathematician	Philosopher
0	1	2	3	4	5	6	7	8	9	10

Values

Hashing Algorithm

- Calculation applied to a key to transform it into an address
- For numeric keys, divide the key by the number of available addresses, n , and take the remainder

$$\text{address} = \text{key} \text{ Mod } n$$

- For alphanumeric keys, divide the sum of ASCII codes in a key by the number of available addresses, n , and take the remainder
- Folding method divides key into equal parts then adds the parts together
 - The telephone number 01452 8345654, becomes $01 + 45 + 28 + 34 + 56 + 54 = 218$
 - Depending on size of table, may then divide by some constant and take remainder

Collisions



Collisions

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

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

Collisions

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					
0	1	2	3	4	5	6	7	8	9	10

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: 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
Len	L	76	e	101	n	110	287	1

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

Collisions: Linear Probing

Find Rae $280 \bmod 11 = 5$

myData = Array(5)

Bea	Tim	Len	Moe	Mia	Zoe	Sue	Lou	Rae	Max	Tod
0	1	2	3	4	5	6	7	8	9	10

Collisions

More data → more probability for collision

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

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

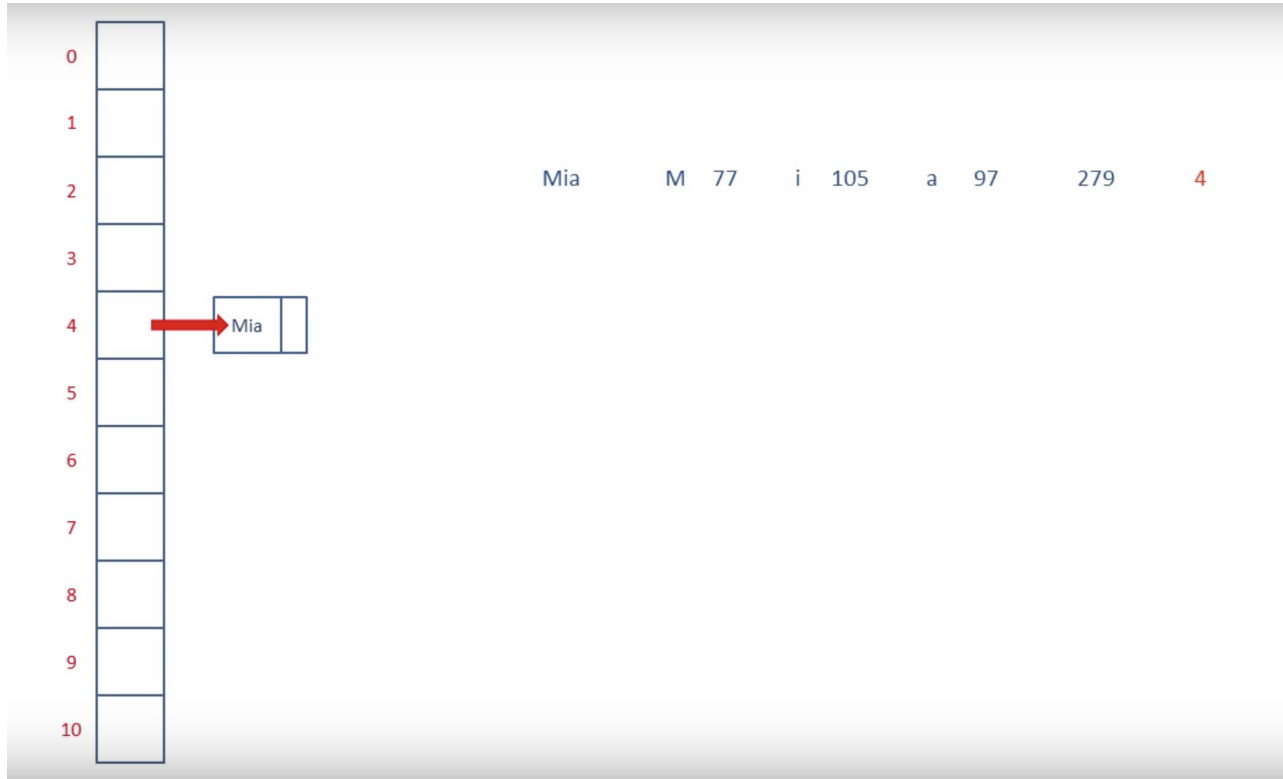
Bea	Tim	Len	Moe	Mia	Zoe	Sue	Lou	Rae	Max	Tod
0	1	2	3	4	5	6	7	8	9	10

Resize the array based on certain threshold of *Load Factor*

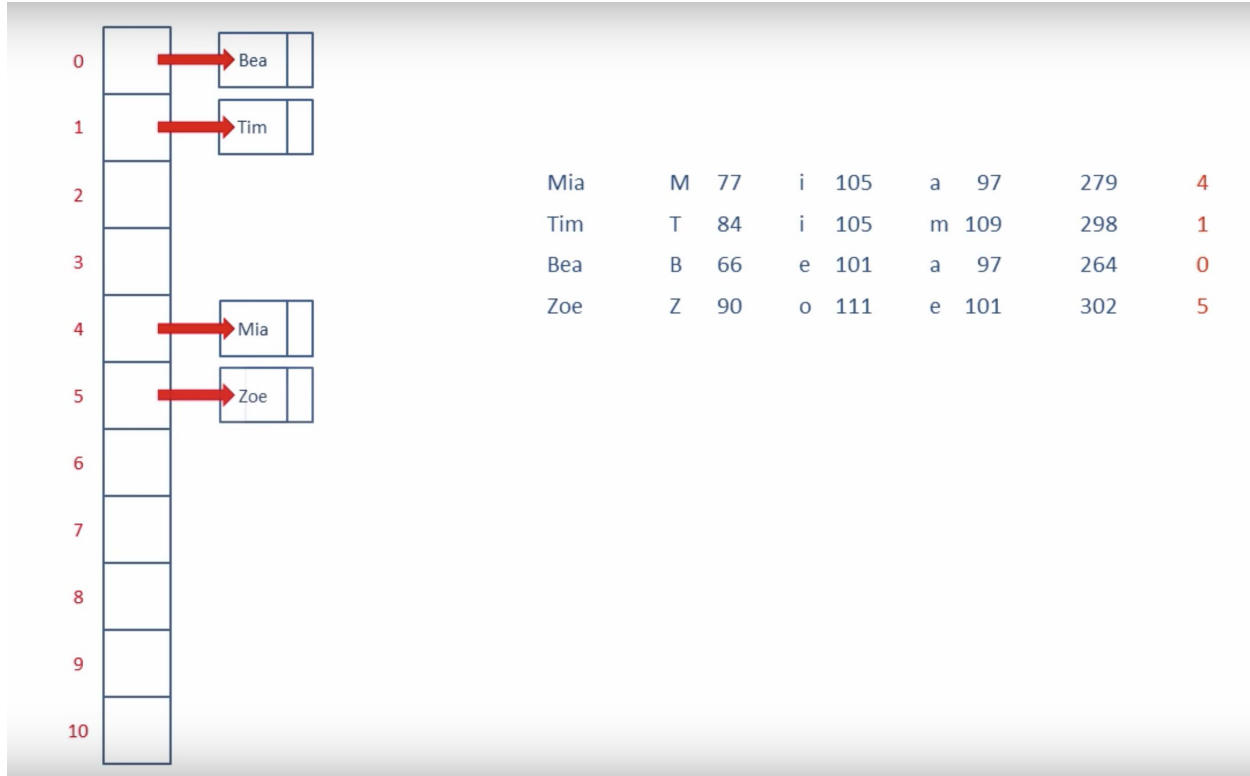
Collisions: Chaining



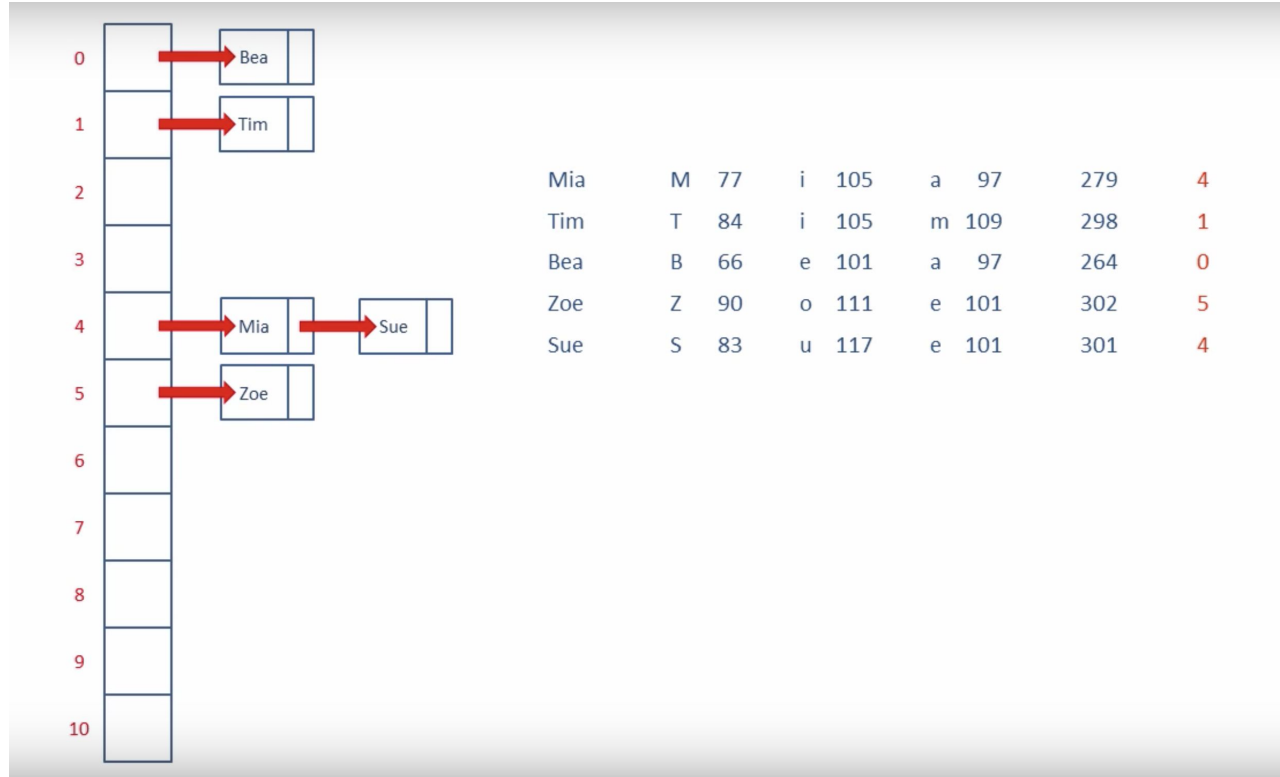
Collisions: Chaining



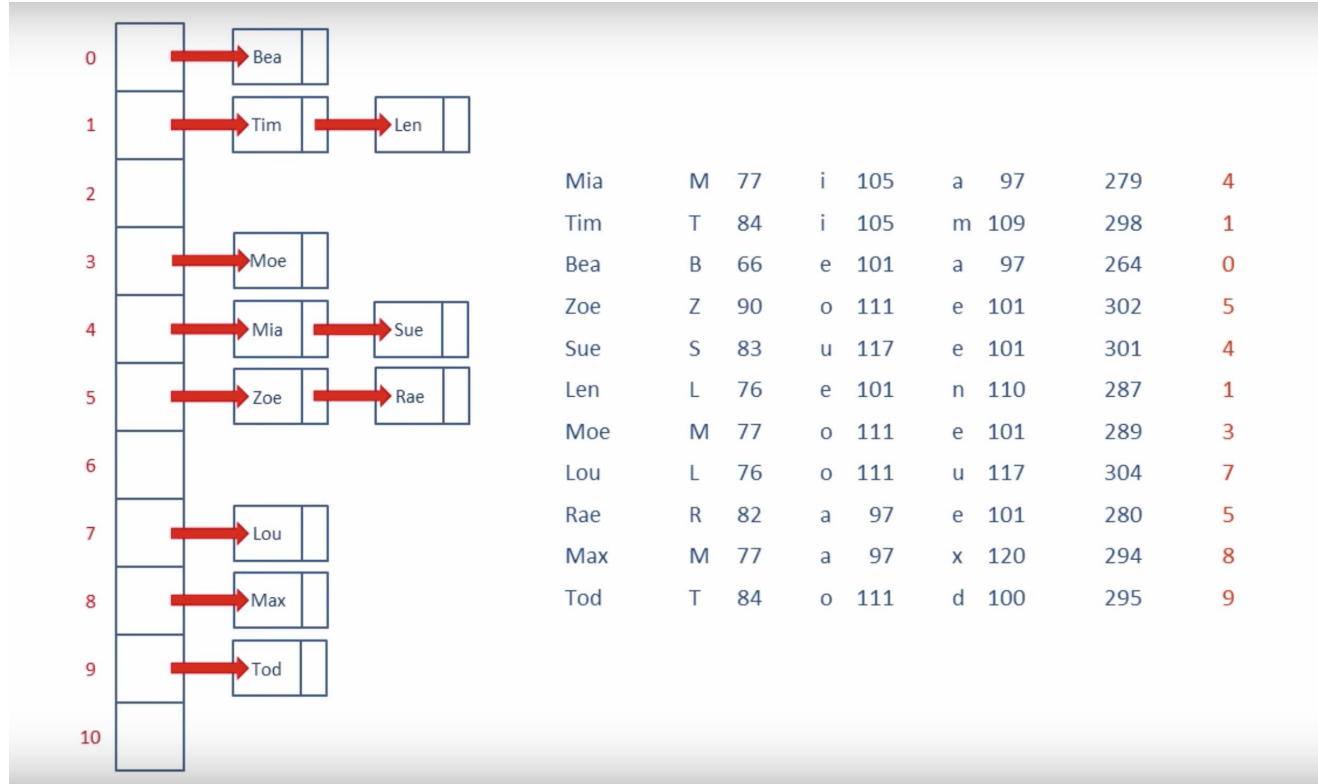
Collisions: Chaining



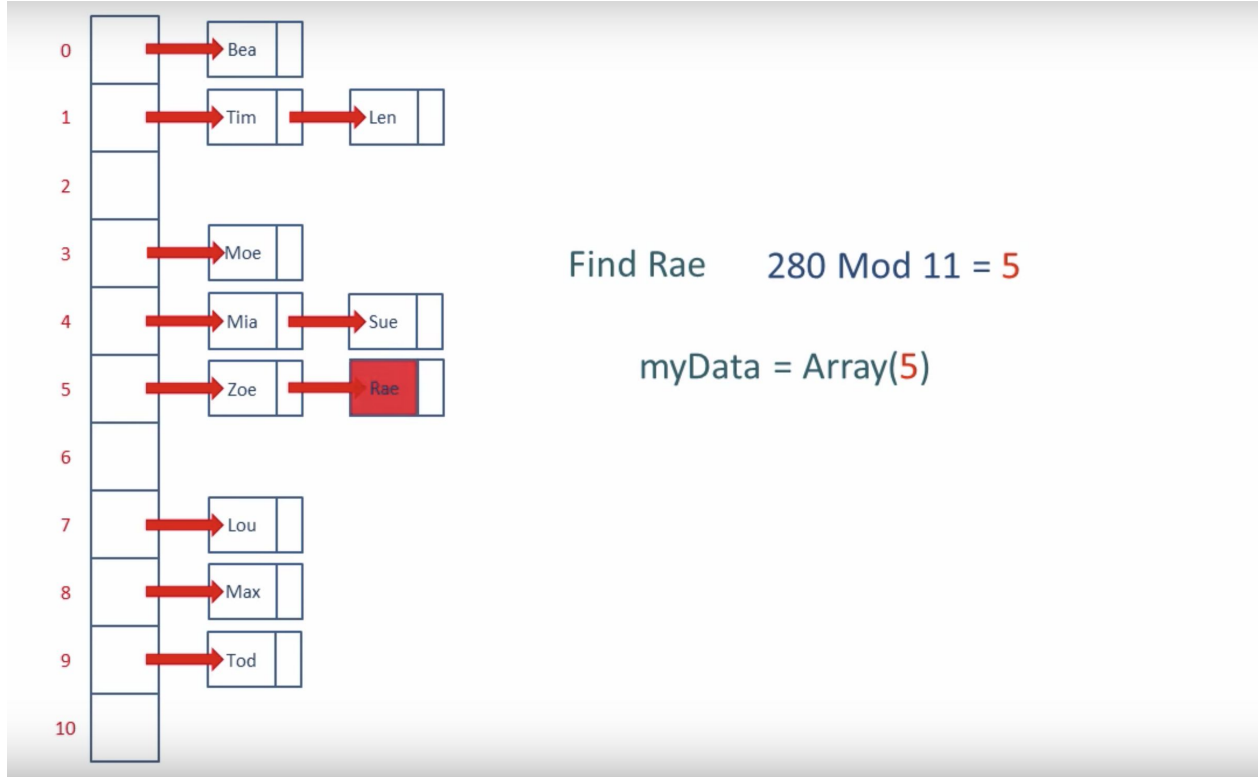
Collisions: Chaining



Collisions: Chaining



Collisions: Chaining



Collison Resolution

- Open addressing
 - Linear probing
 - Plus 3 rehash
 - Quadratic probing (*failed attempts*)²
 - Double hashing
- Closed addressing

How probable are collisions?

Highly Probable!

Birthday paradox:

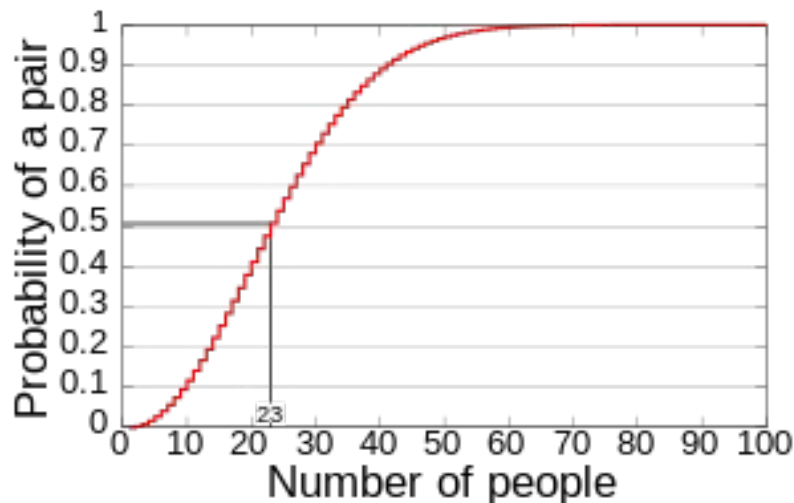
Given a room of 'n' people, at least how many people you require to find among them any two people with matching birthdays with 50% probability?

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

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

How probable are collisions?

Birthday Paradox



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

Objectives of Hash Function

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

Next Lecture

Implementing dictionary using Chaining

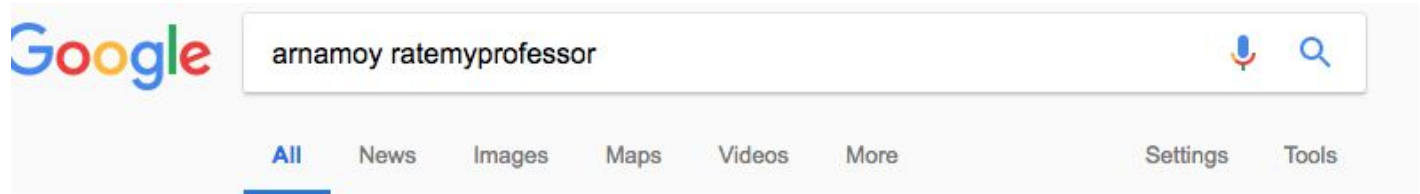
Course Evaluations

Check your e-mail for a link to your evaluations, or log-in to www.portal.utoronto.ca and click the *Course Evals* tab!



CENTRE FOR TEACHING SUPPORT & INNOVATION

Google Arnamoy ratemyprofessor



About 6 results (0.20 seconds)

[Add A Review for Professor Arnamoy Bhattacharyya from University of ...](http://www.ratemyprofessors.com/AddRating.jsp?tid=2325582)

www.ratemyprofessors.com/AddRating.jsp?tid=2325582 ▼

Create review for **Arnamoy** Bhattacharyya from University of Toronto - St. George Campus.

Developing our own Python Dictionary

Onto pycharm