COMP26120 Algorithms and Complexity
Topic 2: Data Structures

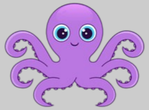
Introducing Hash Maps

Dr. Thomas Carroll
thomas.carroll@manchester.ac.uk
All information on Blackboard

Learning Outcomes

- Understand the ideas of:
  - Hash Map
  - Hash Function
  - Collisions

Where should we go?

0	
1	
2	
3	

Name	Number	%4
Unicorn	21	**1**
Dog	4	**0**
Tiger	20	**0**
Elephant	5	**1**
Octopus	15	**3**

Lookup Table

```
init:
    A = n-sized aray

get(k):
    return A[k]

put(k,v):
    A[k] = v

remove(k):
    A[k] = NULL
```

```
put(1,A)
put(3,B)
put(15,C) ??
put(65536,D)??
put(-1,E) ??
put(banana,F) ??
```

0	1	2	3	4	5	6	7	8	9
	A		B						

Lookup Table + Hashing

```
init:
    A = n-sized aray

get(k):
    return A[hash(k)]

put(k,v):
    A[hash(k)] = v

remove(k):
    A[hash(k)] = NULL

hash(k):
    //Returns value between 0 and n
    return k%n
```

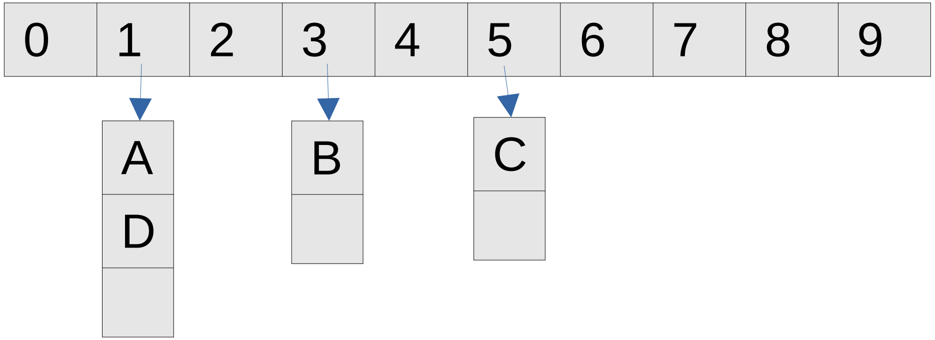
```
put(1,A)
put(3,B)
put(15,C)
put(21,D)??
put(31,E) ??
```

0	1	2	3	4	5	6	7	8	9
	A		B		C				

Lookup Table + Hashing + Separate Chaining

```
init:
    A = n-sized aray

get(k):
    return A[hash(k)].find(k)

put(k,v):
    A[hash(k)].add(k,v)

remove(k):
    A[hash(k)].remove(k)

hash(k):
    //Returns value between 0 and n
    return k%n
```

```
put(1,A)
put(3,B)
put(15,C)
put(21,D)
```



0	1	2	3	4	5	6	7	8	9

What Else?

- Hashing
  - Key → Index
- Open Addressing
  - Dealing with collisions
- Load Factor / Rehashing
  - When and how to expand the array?
- Complexity
  - Can we argue that lookup and insertion takes constant time?