

I ILLINOIS

Hash Table

Learning Objectives

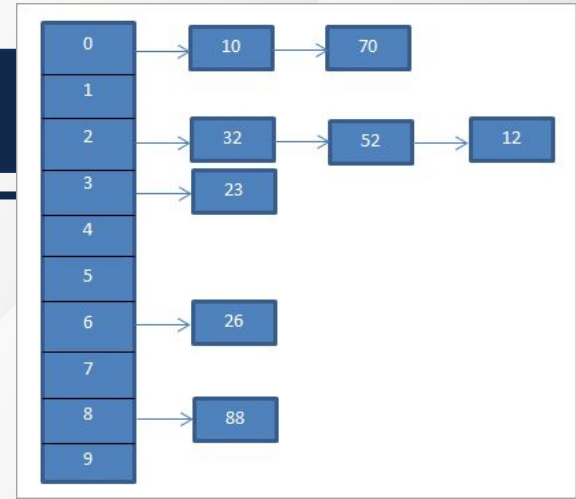
1. Understand how collisions are handled with open hashing
2. Understand how insert, remove, and find are implemented for dictionaries with open hashing



Types of Hash Tables

Open Hashing (Closed Address) (Separate Chaining)

- Simpler, decent performance
- HashTable points to external storage
- Index tells you where the data is



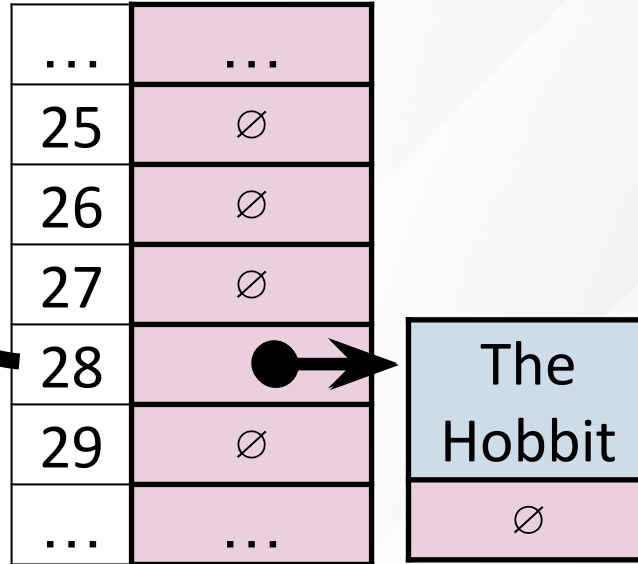
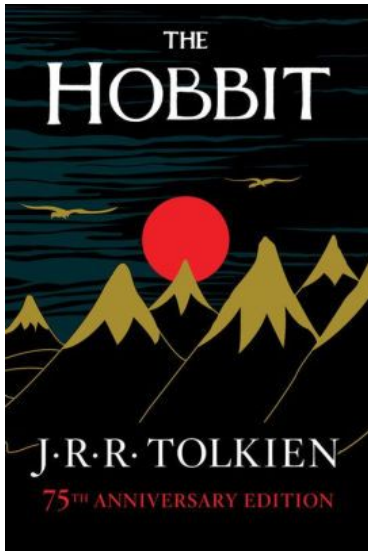
Closed Hashing (Open Address)

- Complex, high performance
- HashTable is an array
- Location of data depends on contents of hash table

--	--	--	--	--

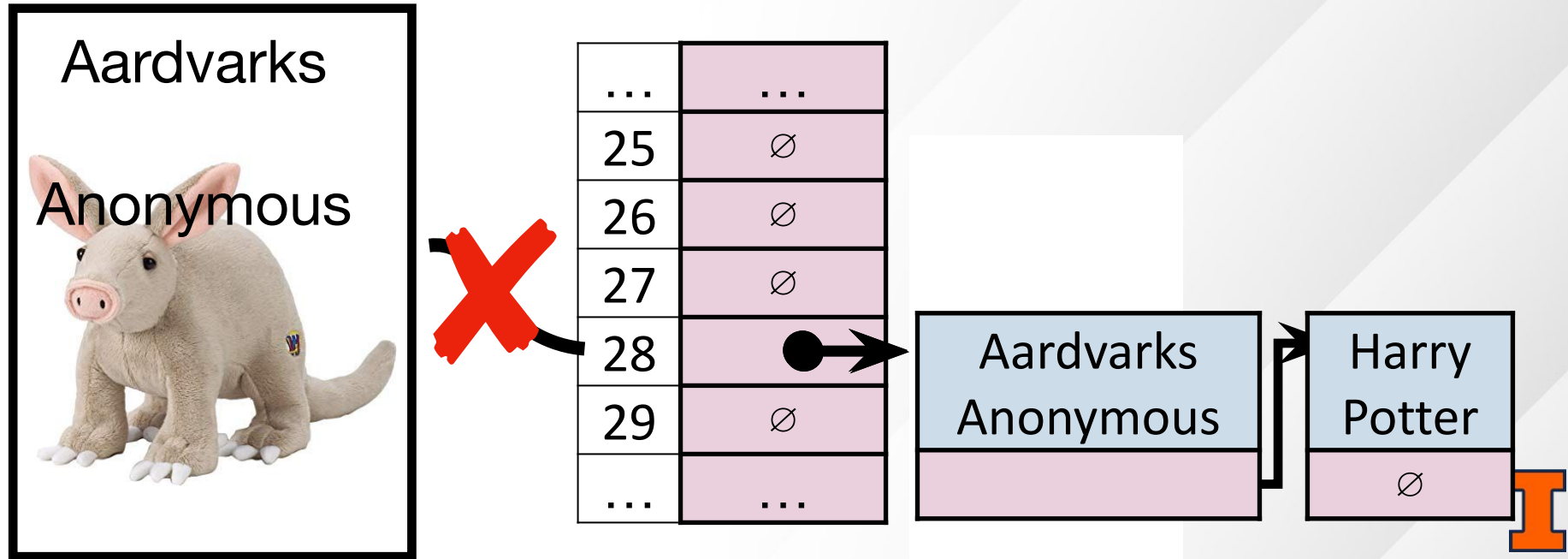
Open Hashing

In an *open hashing* scheme, key-value pairs are stored externally (for example as a linked list).



Hash Collisions (Open Hashing)

A *hash collision* in an open hashing scheme can be resolved by _____. This is called *separate chaining*.



Insertion (Separate Chaining)

Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7

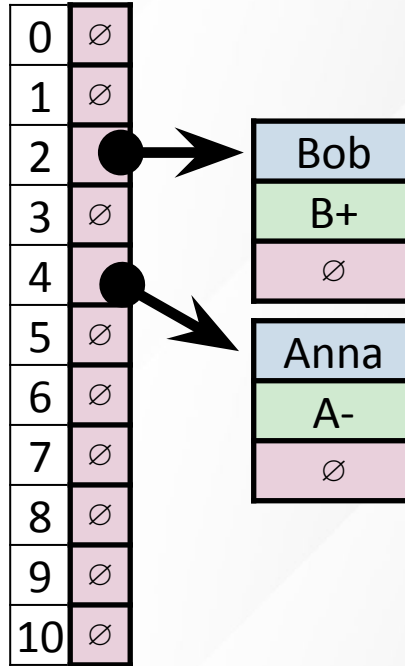
0	∅
1	∅
2	∅
3	∅
4	∅
5	∅
6	∅
7	∅
8	∅
9	∅
10	∅

_insert("Bob")

_insert("Anna")

Insertion (Separate Chaining)

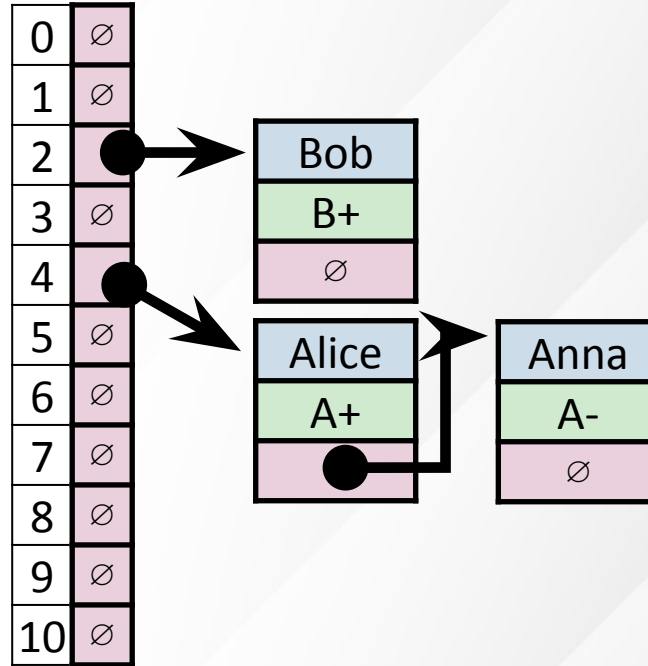
Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7



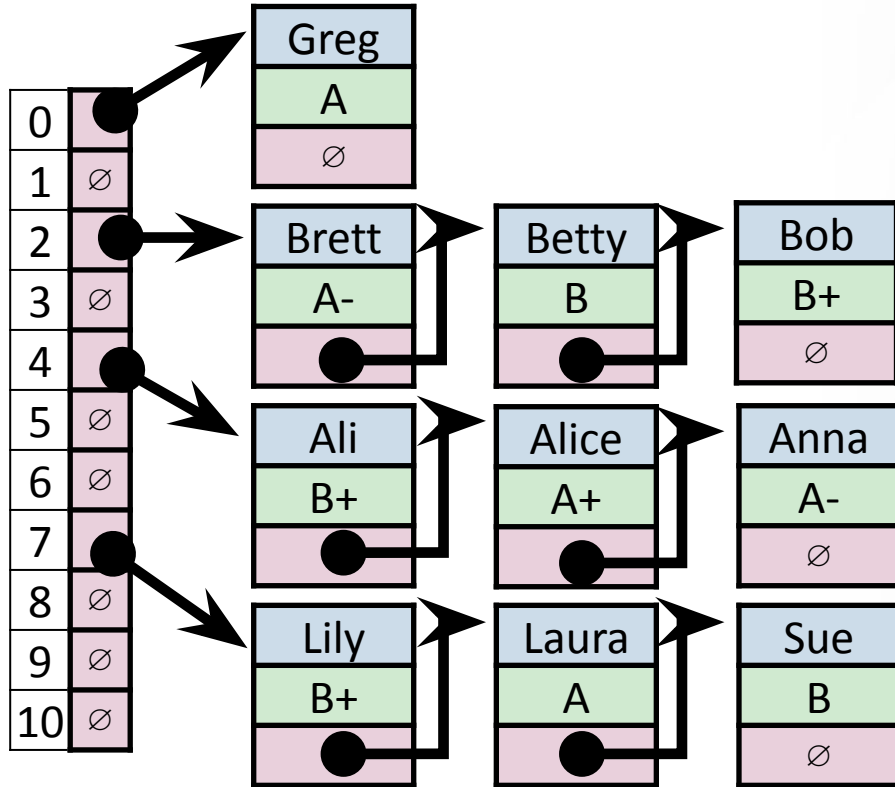
_insert("Alice")

Insertion (Separate Chaining)

Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7

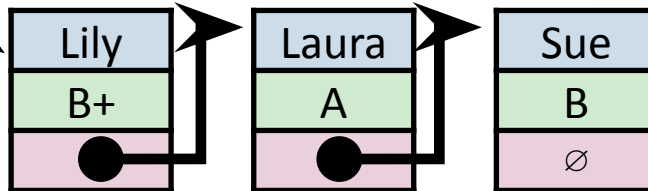


Insertion (Separate Chaining)



Find (Separate Chaining)

0	∅
1	∅
2	∅
3	∅
4	∅
5	∅
6	∅
7	●
8	∅
9	∅
10	∅

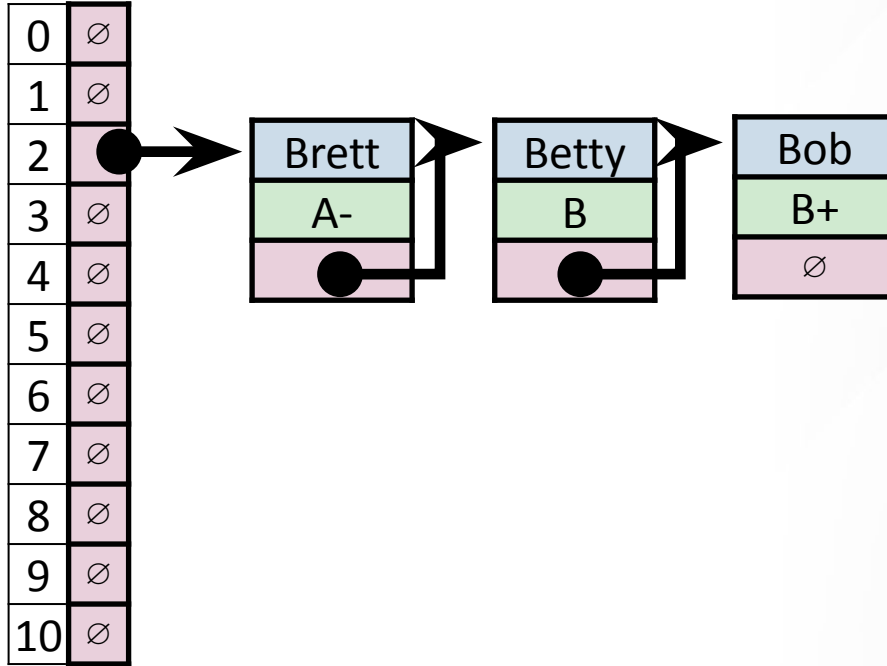


`_find("Sue")`

Key	Hash
Sue	7



Remove (Separate Chaining)



`_remove("Betty")`

Key	Hash
Betty	2

Hash Table (Separate Chaining)

For hash table of size m and n elements:

Find runs in: _____

Insert runs in: _____

Remove runs in: _____

