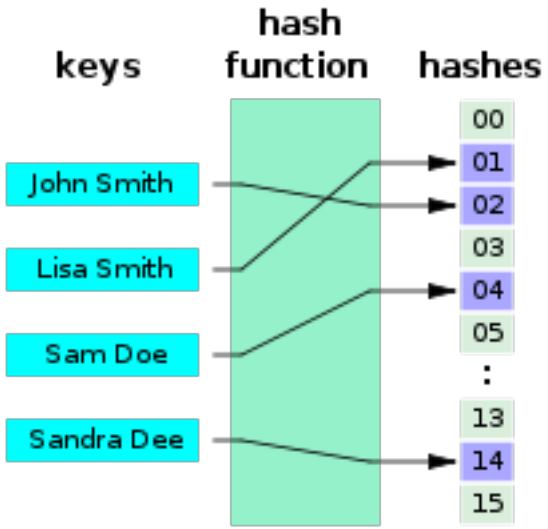
CS 2230
CS II: Data structures

Meeting 29: Hashing

Brandon Myers

University of Iowa

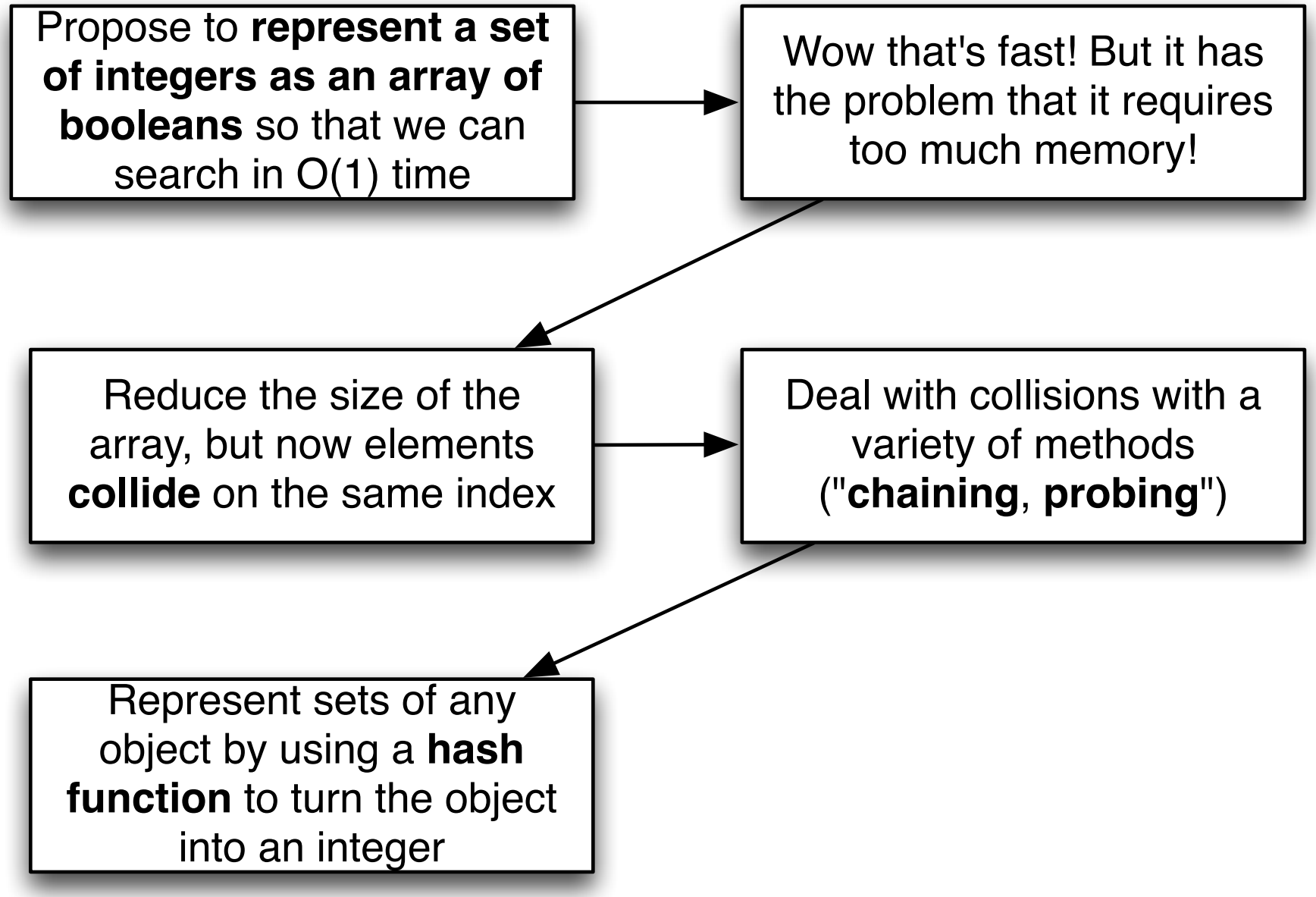


https://en.wikipedia.org/wiki/Hash_function

Today's learning objectives

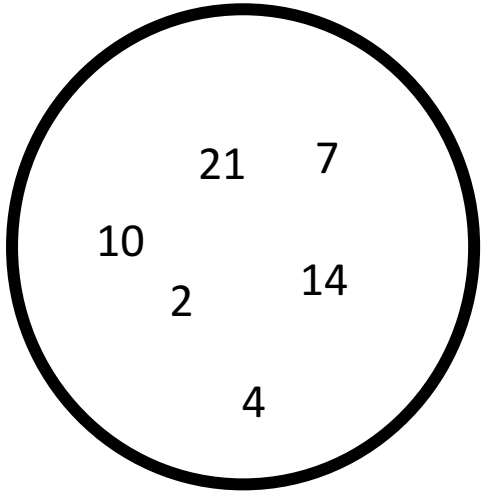
- Identify various data structures to implement a Set
- Calculate the memory usage of hashing data structures
- Execute the Set methods for various hash set implementations, including when there are collisions
- Identify important properties of hash codes

Roadmap

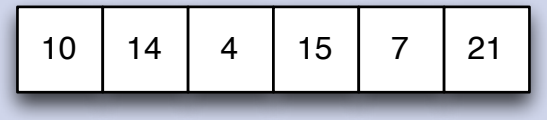
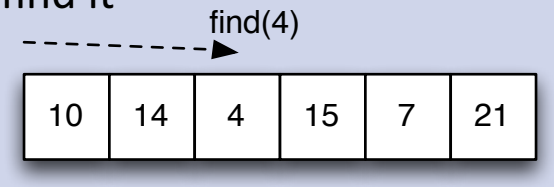
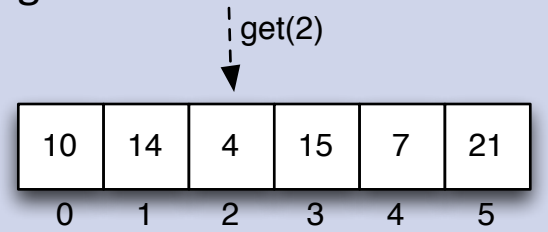

Propose to **represent a set of integers as an array of booleans** so that we can search in O(1) time
Wow that's fast! But it has the problem that it requires too much memory!
Reduce the size of the array, but now elements **collide** on the same index
Deal with collisions with a variety of methods ("**chaining**, **probing**")
Represent sets of any object by using a **hash function** to turn the object into an integer

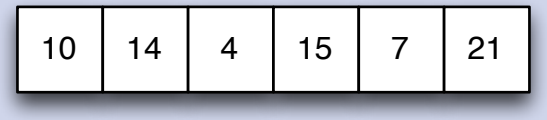
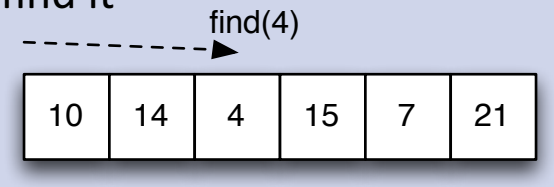
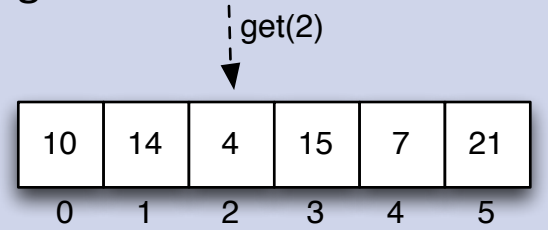
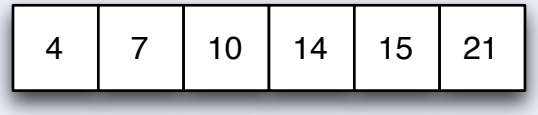
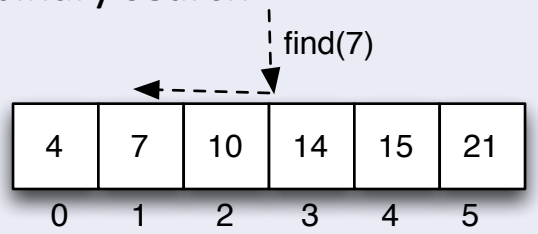
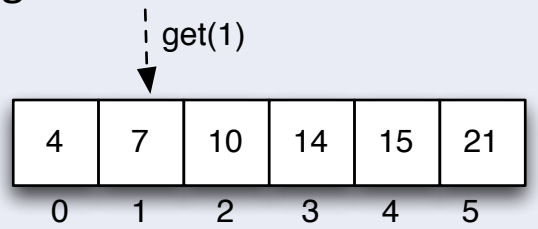
Identify various data structures to implement a Set

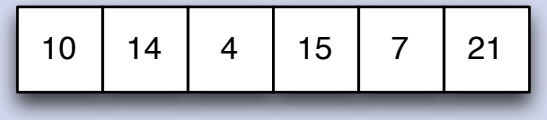
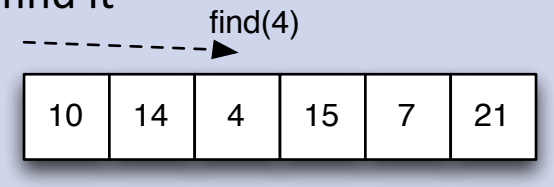
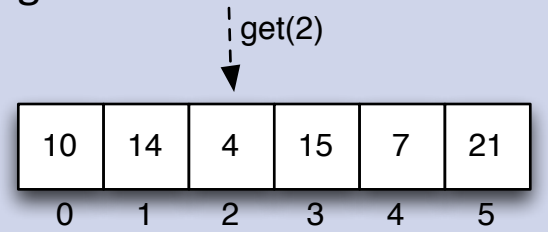
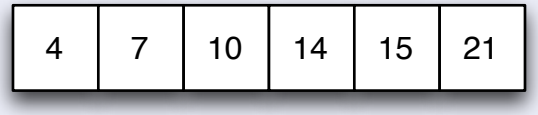
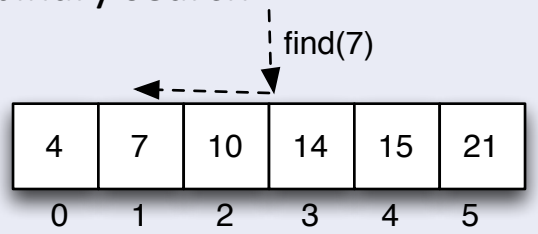
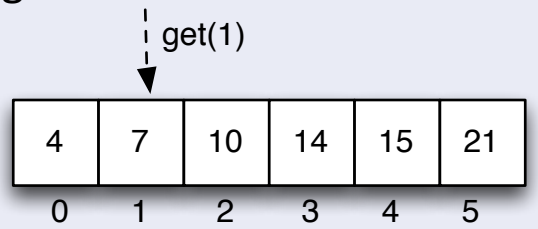
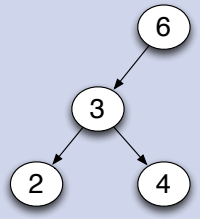
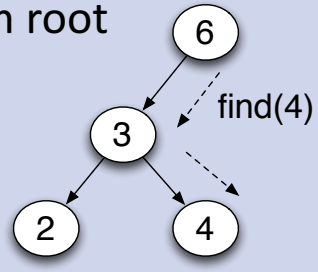
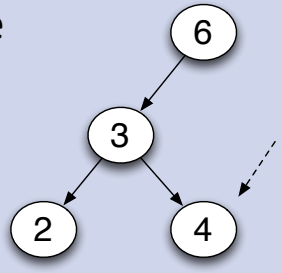
What are ways we can represent a Set of integers?

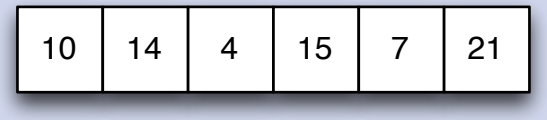
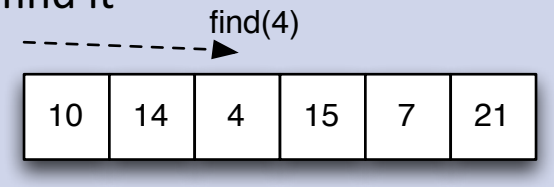
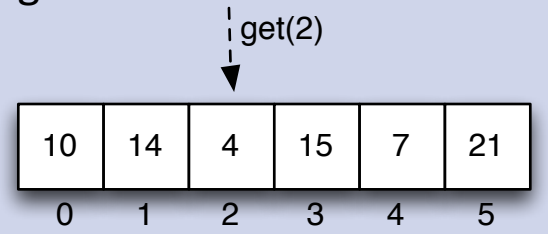
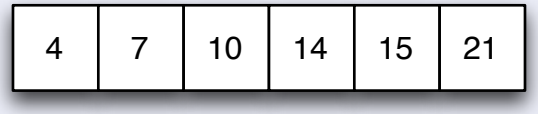
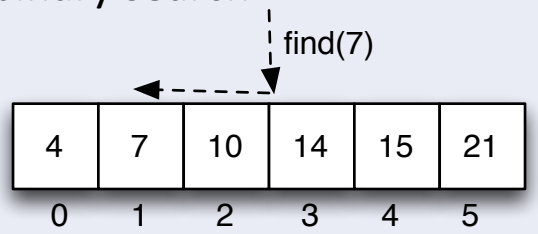
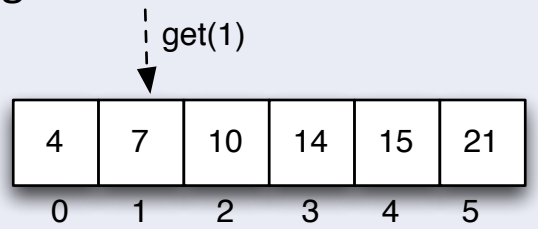
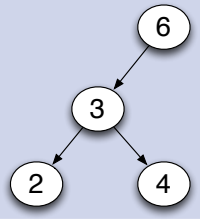
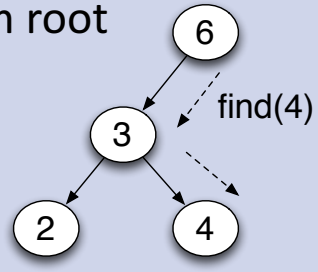
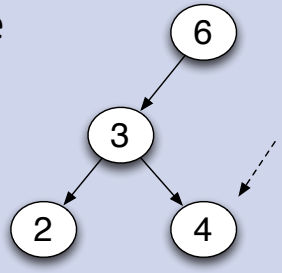
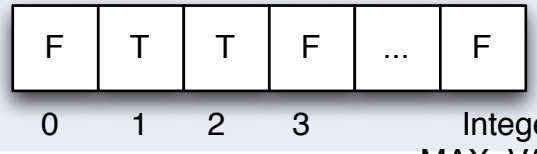
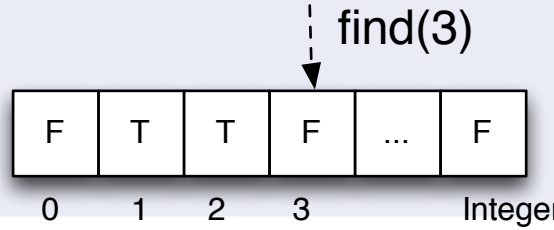
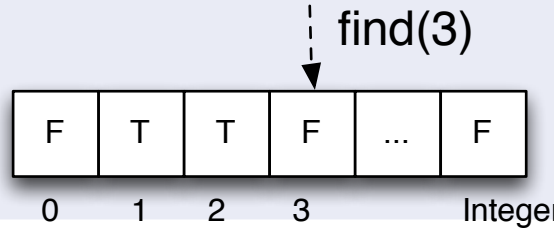


https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+

Data structure	how to search for a specific value	if you know where it is stored (e.g., index or reference)
unsorted array of integers 	search from start until we find it 	go to the index 

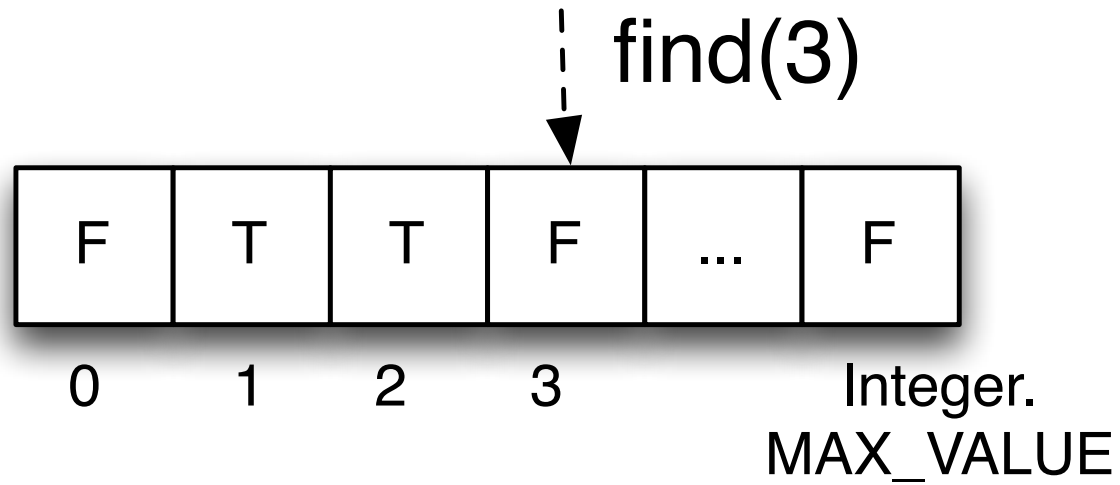
Data structure	how to search for a specific value	if you know where it is stored (e.g., index or reference)
unsorted array of integers 	search from start until we find it 	go to the index 
sorted array of integers 	binary search 	go to the index 

Data structure	how to search for a specific value	if you know where it is stored (e.g., index or reference)
unsorted array of integers 	search from start until we find it 	go to the index 
sorted array of integers 	binary search 	go to the index 
binary search tree of integers 	search from root 	go to the node 

Data structure	how to search for a specific value	if you know where it is stored (e.g., index or reference)
unsorted array of integers 	search from start until we find it 	go to the index 
sorted array of integers 	binary search 	go to the index 
binary search tree of integers 	search from root 	go to the node 
huge array of booleans (true means the value is in the set) 	use the value as an index 	same as search 

```
boolean[] set = new boolean[Integer.MAX_INT+1];
set[1] = true; // add 1
set[2] = true; // add 2
```

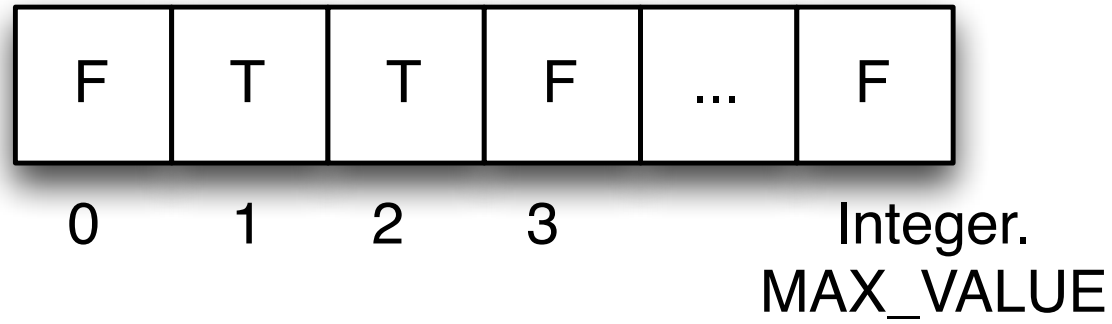
This data structure is great! Find any value in O(1) time!



Problems?

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+

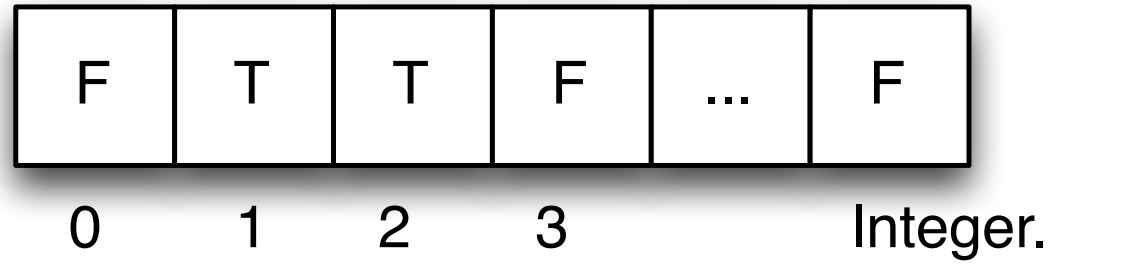
Calculate the memory usage of hashing data structures



Let $M(n)$ be the amount of ***memory*** this Set uses, where n=number of elements in the Set. Which is true and the best bound?

a) $M(n) \in O(1)$
b) $M(n) \in O(logn)$
c) $M(n) \in O(n)$
d) $M(n) \in O(Integer.MAX\_INT)$
e) $M(n) \in O(n * Integer.MAX\_INT)$

For example…



`Integer.MAX_VALUE`$= 2^{31} - 1$
boolean data type is 1 to 2 bytes

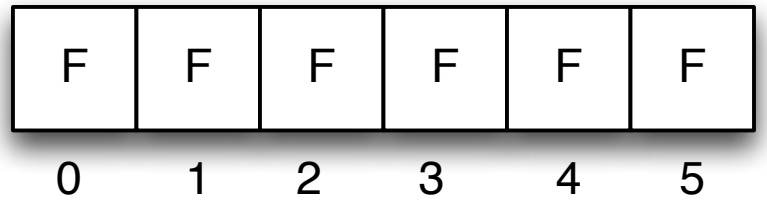
2^{31}-1 * 2 bytes = ~4GB even if your set is nearly empty!

If you are clever and represent the boolean as 1 bit each (0=false, 1=true) then you can get down to 268MB

Even if 268MB fits in your computer's RAM, reality bites you: if your elements are uniformly randomly distributed across those 268MB then the elements of your set won't all be in your computer's fast cache memory, which has a capacity in the 100s of KB (take CS:2630 to learn more!)

Fixing the memory problem

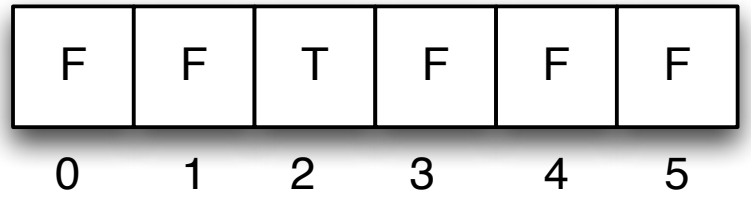
Limit the array to a smaller capacity, say 6



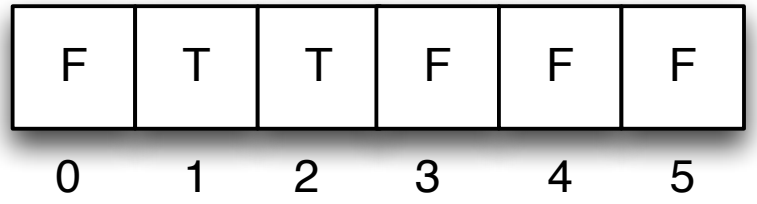
how to add(i): mark true at index $i \ mod \ capacity$

(bonus: we can also store negative integers now)

add(2)

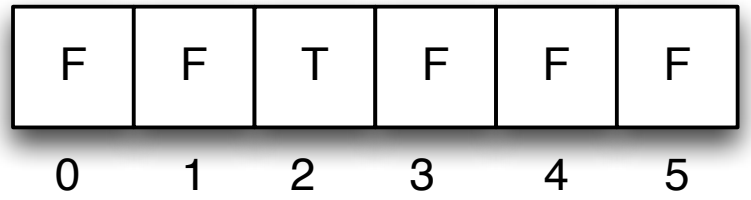


add(7)

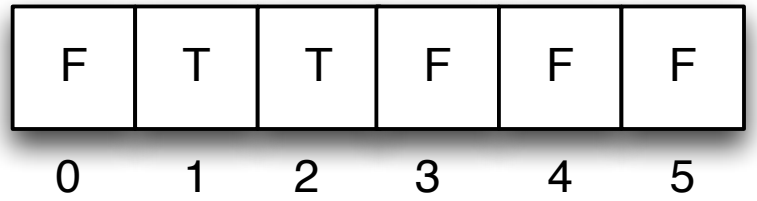


a new problem! It looks like 1 is in the set (and 13,19,25, ...) even though we only added 7

add(2)



add(7)

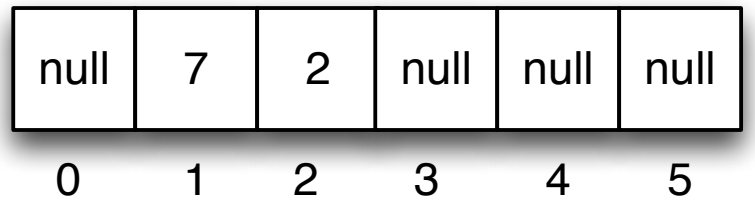


a new problem! It looks like 1 is in the set (and 14,21,28, ...) even though we only added 7

Since many values (1,7,13,19,25,...) map to index 1, we need to keep track of *which* key is stored there

We'll have the array store Integers, where null means the bucket is empty and a non-null value is the key stored there

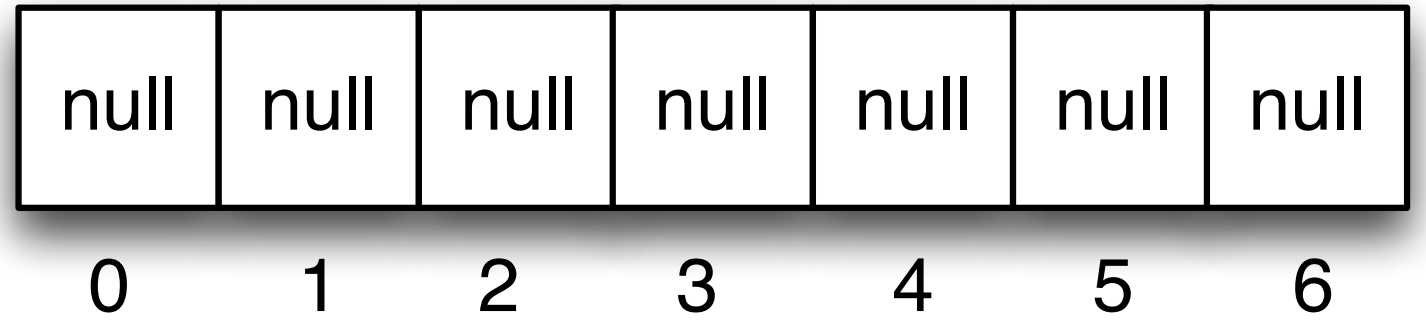
```
Integer[] set = new Integer[6]; // capacity=6
set[2 % 6] = 2;
set[7 % 6] = 7;
```



% means mod

Execute the Set methods for various hash set implementations

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+



Suppose our set is initially empty as above. What will it look like after the following elements are added? -1, 19, 17, 21, and 8

a)

-1	19	17	21	8	null	null

b)

-1	8	17	19	21	null	null

c)

null	19	8	21	null	17	-1

d)

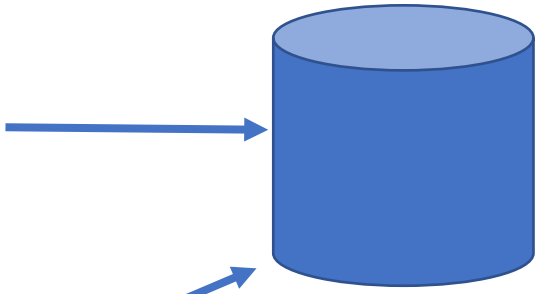
21	8	null	17	null	19	-1

Learning objectives for Final project

- Design, implement, and test an application based on a written specification
- Choose appropriate ADTs and efficient data structures for various tasks
- Use version control to collaborate on a coding project with another person

Final project:
Semantic similarity of words

3 May. Bistritz.--Left Munich at 8:35 P. M., on 1st May, arriving at Vienna early next morning; should have arrived at 6:46, but train was an hour late. Buda-Pesth seems a wonderful place, from the glimpse which I got of it from the train and the little I could walk through the streets....



3 similar words to “time”?

come, 0.6202651310028829
sleep, 0.613304123466795
time, 0.6082294707042364

our definition of semantic meaning: the number of times a word appears with other words in the same sentence. Each word becomes a vector.

I am a sick man. I am a spiteful man. I am an unattractive man. I believe my liver is diseased. However, I know nothing at all about my disease, and do not know for certain what ails me.

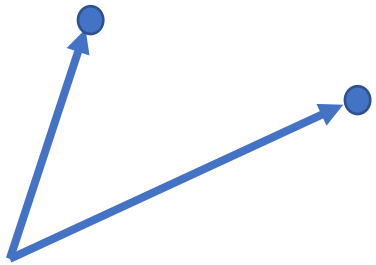
The word “man” appears in the first three sentences. Its semantic descriptor vector would be:

[I=3, am=3, a=2, sick=1, man=0, spiteful=1, an=1, unattractive=1, believe=0, my=0, liver=0, is=0, diseased=0, However=0, know=0, nothing=0, at=0, all=0, about=0, disease=0, and=0, do=0, not=0, for=0, certain=0, what=0, ails=0, me=0]

The word “liver” occurs in the fourth sentence, so its semantic descriptor vector is:

[I=1, am=0, a=0, sick=0, man=0, spiteful=0, an=0, unattractive=0, believe=1, my=1, liver=0, is=1, diseased=1, However=0, know=0, nothing=0, at=0, all=0, about=0, disease=0, and=0, do=0, not=0, for=0, certain=0, what=0, ails=0, me=0]

various measures of similarity of two vectors
cosine similarity
negative Euclidean distance
negative Euclidean distance of norms

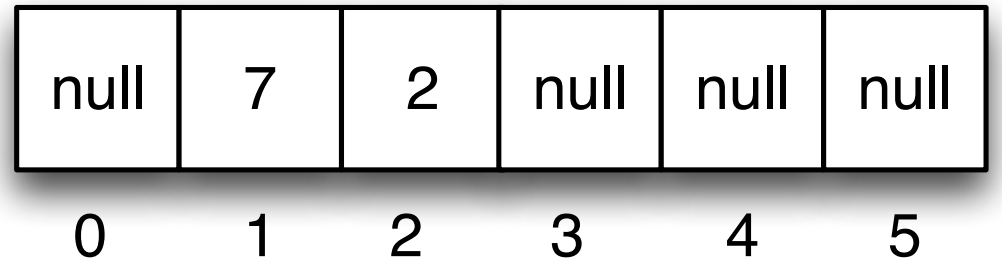


use these measures to answer queries about the words in a text

Project due dates

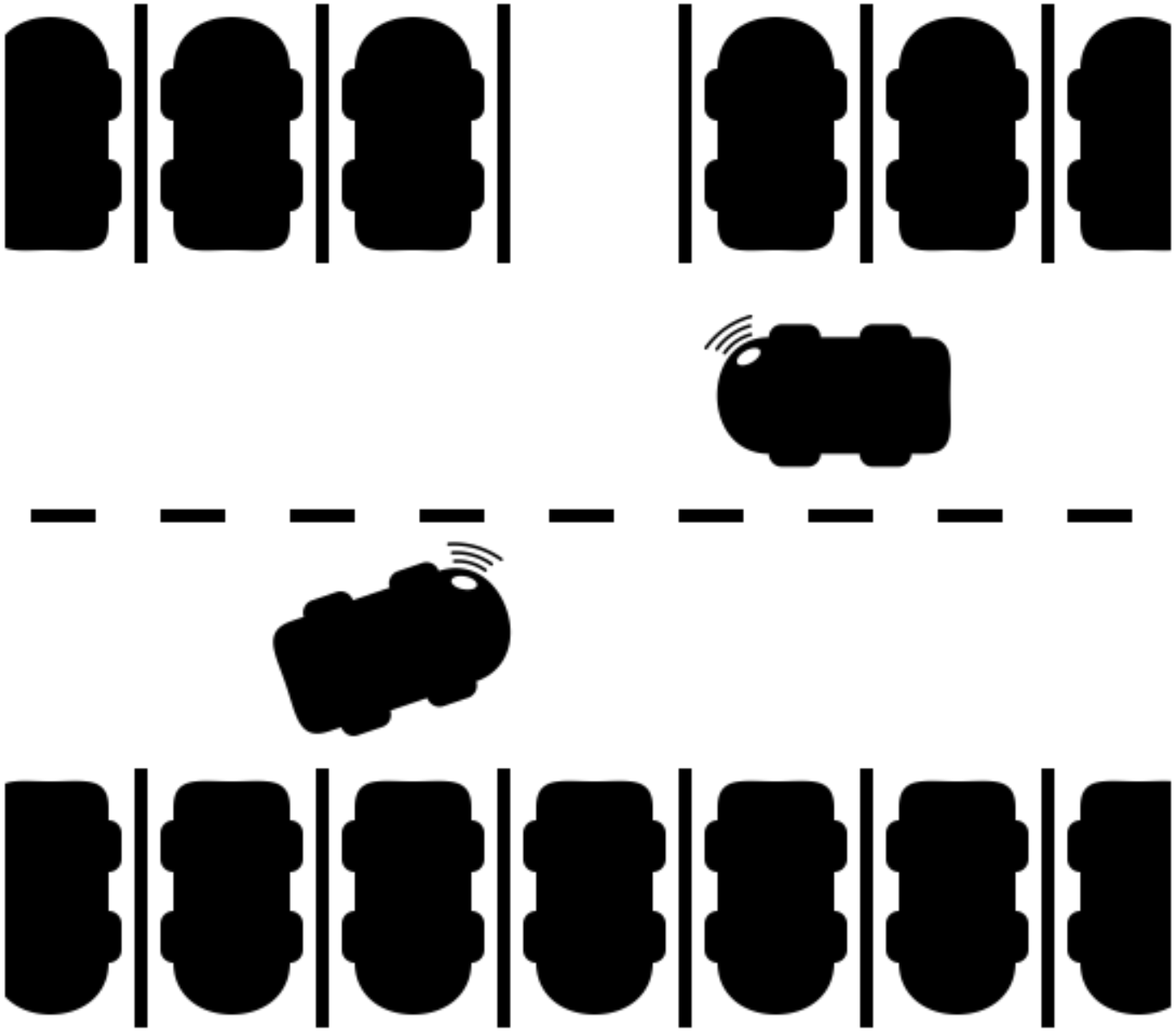
- Nov 17, 11:59pm: Milestone 1 in GitHub (no slip days)
  - Finished Part 1
  - PROGRESS_REPORT_NOV17.txt
- Nov 29, 11:59pm: Milestone 2 in GitHub (no slip days)
  - Finished Part 3, and initial draft of Part 4's written answers
  - PROGRESS_REPORT_NOV29.txt
- Dec 6, 11:59pm: Final version in GitHub (up to 2 slip days if at least 1 partner has them)
  - Finished all Parts

Collisions!



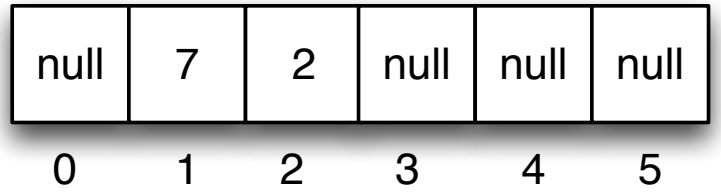
add(13) // 13 % 6 = 1

uhoh...



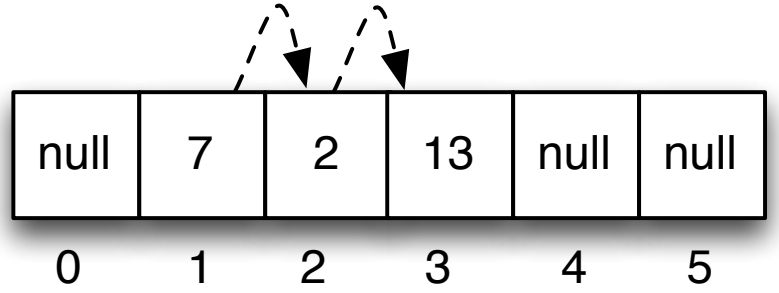
You know that feeling...
when someone takes your parking spot

Dealing with collisions

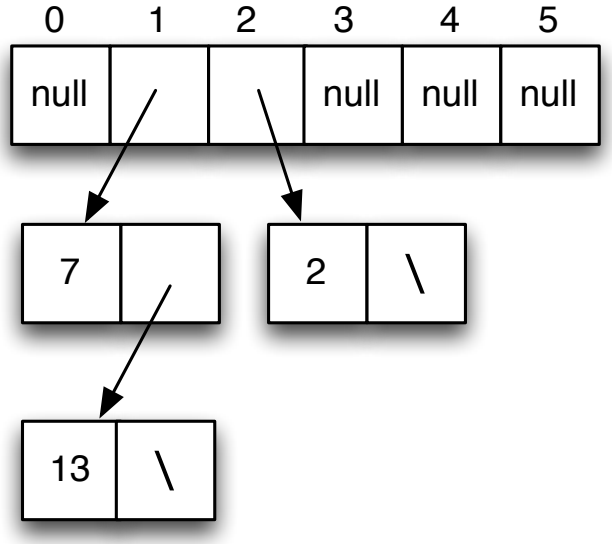


add(13) // 13 % 6 = 1

Linear probing
Go to the next spot until you find an opening



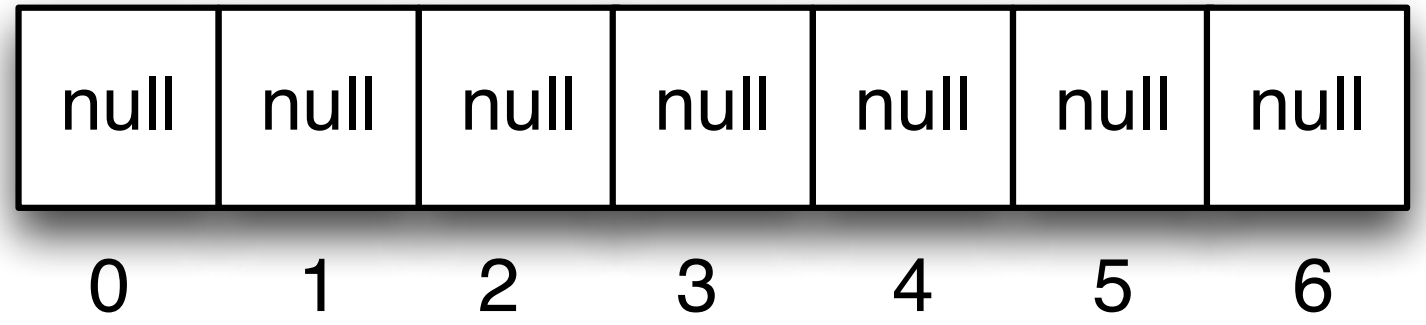
Chaining
Each bucket is a linked list of elements stored there



...and other techniques!

Execute the Set methods for various hash set implementations, including when there are collisions

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+



Suppose our set is initially empty as above. What will it look like after the following elements are added, assuming we use **linear probing**? 9, 18, 23, 17

a)

null	null	9	23	18	17	null

b)

null	null	23	17	4	null	null

c)

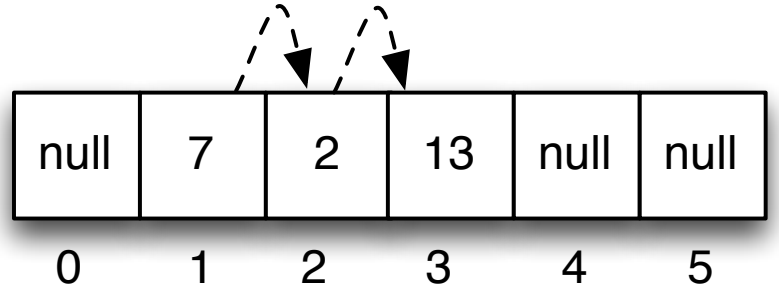
null	null	9	18	23	17	null

d)

null	null	9	null	18	null	null

Execute the Set methods for various hash set implementations, including when there are collisions

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+

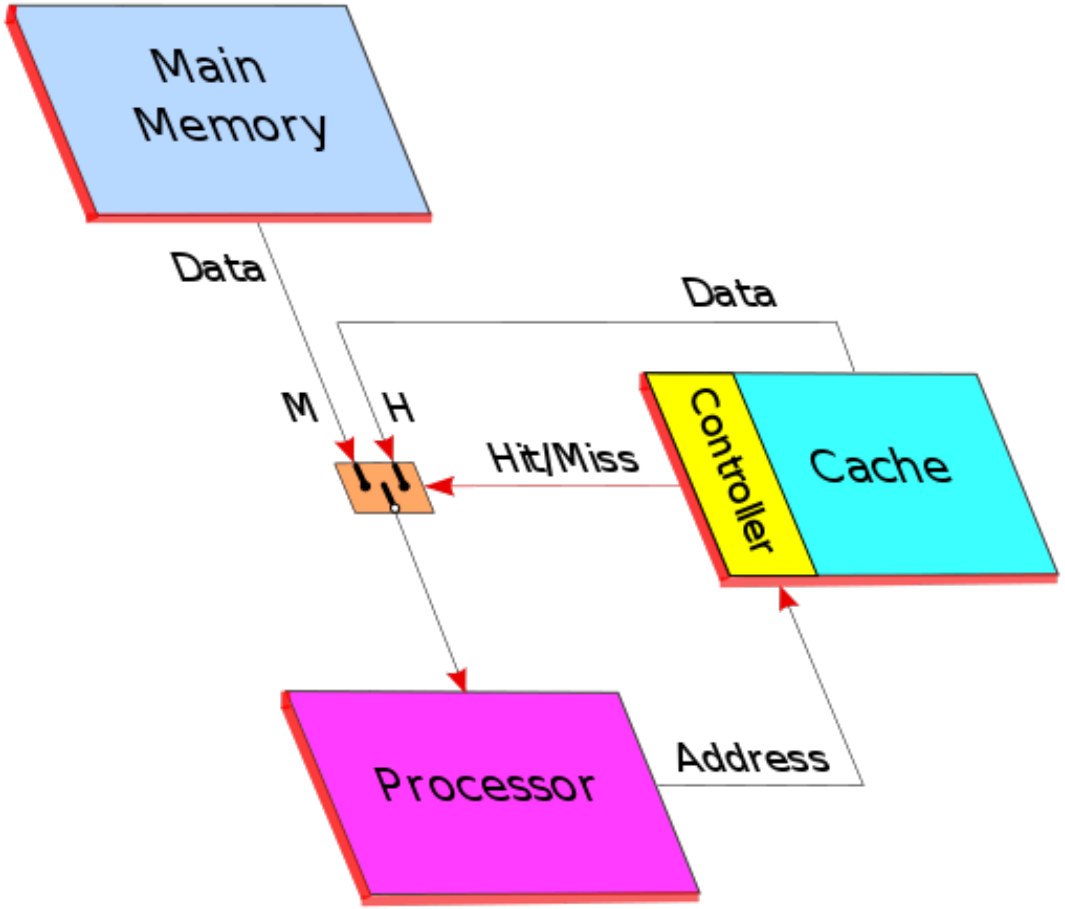


How should we implement remove() if we are using linear probing? (e.g., remove(7))

a) set the the bucket to null
b) Remove the element and move all elements after it left by one space
c) Move all elements after it (up to the next null) left by one space
d) leave a special marker in the bucket that means it is deleted
e) there is no good way to allow remove()

The hash set in your smartphone's processor that you didn't know about

all your data (e.g., running programs, the OS, and their data)



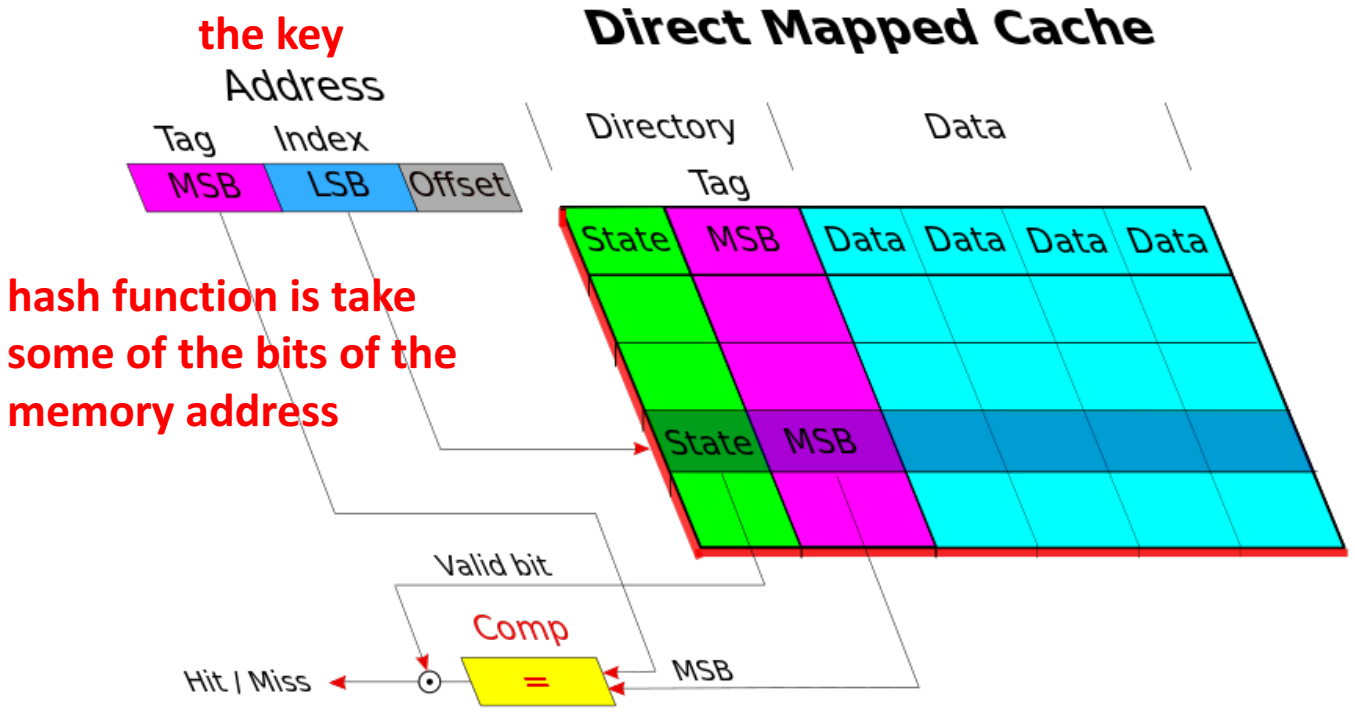
cache stores a subset of your data

it is small but that makes it fast!

The hash set in your smartphone's processor that you didn't know about

The cache is basically a hash set

here's one where each bucket can only hold 1 key

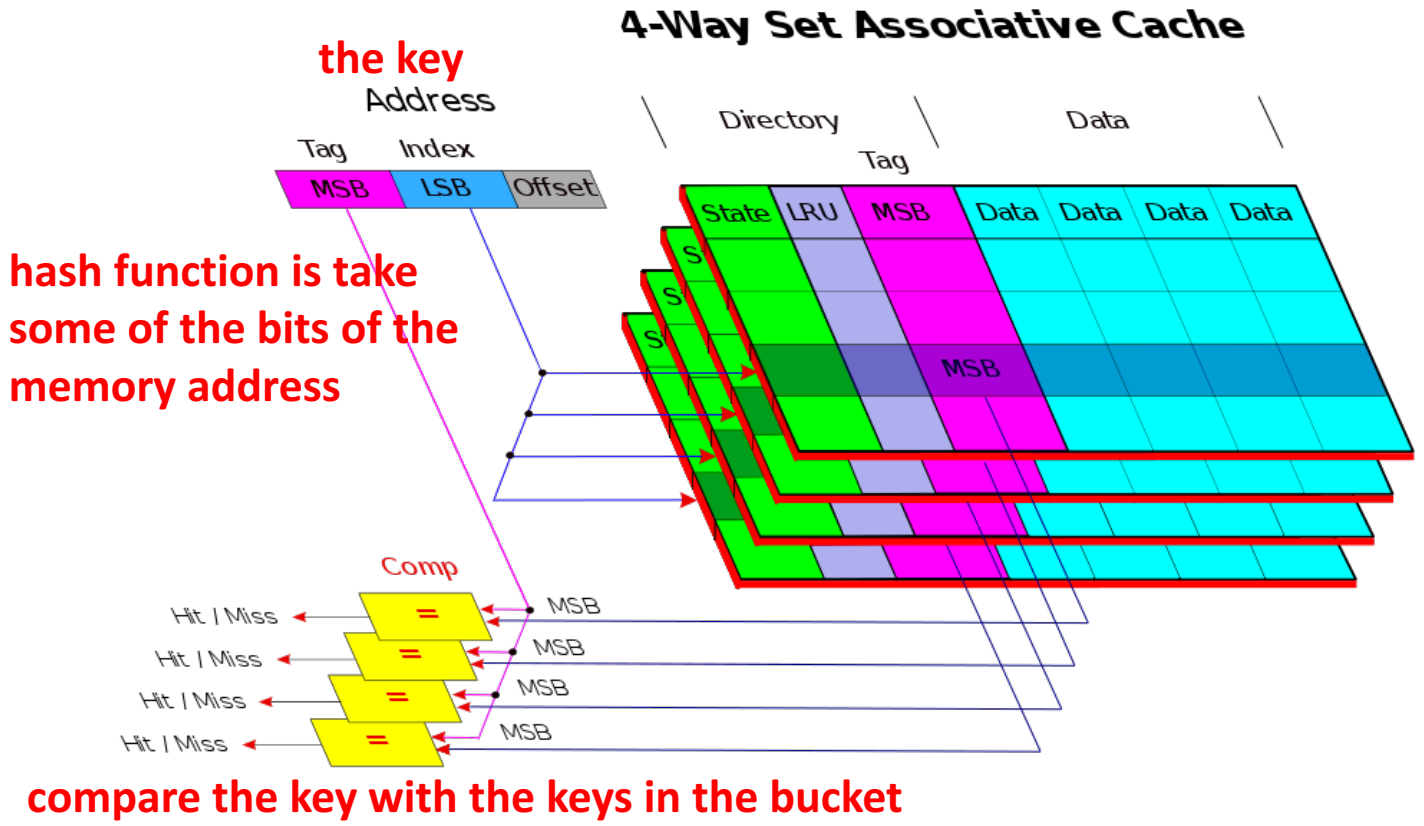


compare the key with the key in the bucket

The hash set in your smartphone's processor that you didn't know about

The cache is basically a hash set

here's one where each bucket can hold up to 4 keys



Putting non-integers into a set

```
String[] set = new String[capacity];
set[???] = "Cat";
```

Where should we put the string “Cat”?

Putting non-integers into a set

```
String[] set = new String[capacity];
set[???] = "Cat";
```

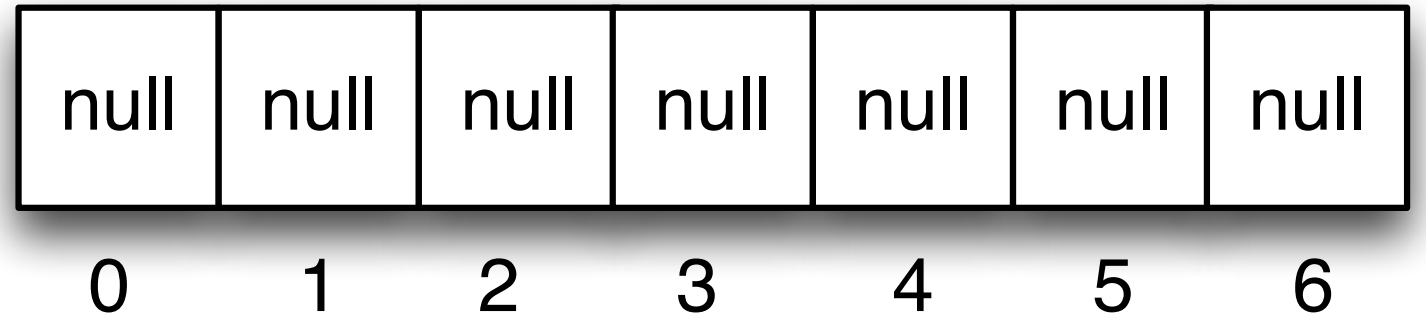
Where should we put the string “Cat”?

use a ***hash function***

a hash function is just any function that turns an object into an integer

Execute the Set methods for various hash set implementations, including when there are collisions

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+



Suppose the hash function for a string is the length

What is the contents after inserting “Cat”, “Dog”, “Froggy”? Assume we use Linear Probing.

a)

null	null	null	“Cat”	“Dog”	”Froggy”	null

b)

“Froggy”	null	null	“Cat”	“Dog”	null	null

c)

“Cat”	”Dog”	”Froggy”	null	null	null	null

d)

null	null	null	“Cat”	“Dog”	null	“Froggy”

a hash function is just any function that turns an object into an integer

for example, Oracle Java distribution's hash function for Strings

```
public class String {
    // a string is stored as an array of
    // "chars" (characters)
    private final char value[];


    // hash function for String
    public int hashCode() {
        int h = hash;
        if (h == 0 && value.length > 0) {
            char val[] = value;


            for (int i = 0; i < value.length; i++) {
                h = 31 * h + val[i];
            }
            hash = h;
        }
        return h;
    }
}
```

A paraphrase of Object.hashCode specification in the Java API

- The general contract of hashCode is:
  - during the same run of your program, hashCode on a specific object must always return the same result
  - o1.equals(o2) ⇒ o1.hashCode() == o2.hashCode()
  - Important to know that

o1.hashCode() == o2.hashCode DOES NOT IMPLY o1.equals(o2)

i.e., it is okay for two different objects to have the same hashCode (and pretty much impossible to avoid)

https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#hashCode()

Identify important properties of hash codes

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+

If I have the following code

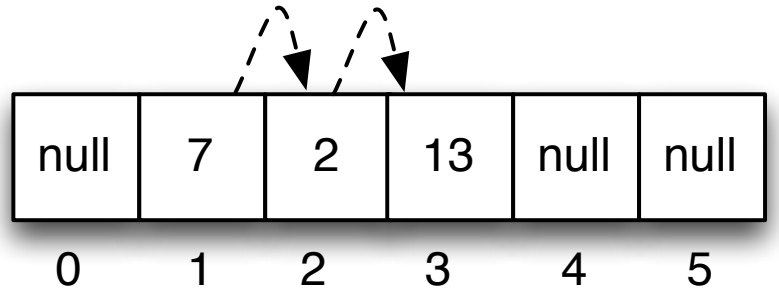
```
Map<Cat, Dog> x = new HashMap<Cat,Dog>();
```

which of the following statements is true

a) If you override Cat.equals you must override Cat.hashCode
b) You must override Dog.equals and Dog.hashCode
c) You must override Cat.equals, Cat.hashCode, Dog.equals, and Dog.hashCode

Running time of successful find?

- Linear probing



  - expected length of a sequence of non-nulls: $O\left(1+\frac{1}{1-\alpha}\right)$ where α is the ***load factor***

  - where $\alpha = \frac{\#\ occupied}{capacity}$ (α is called the ***load factor***)

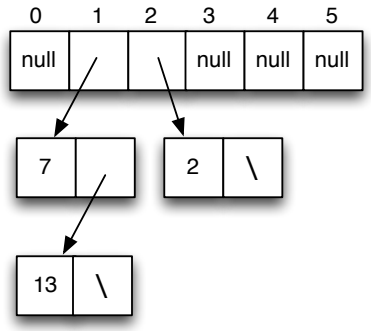
  - worst case: O(n) if the table is allowed to get nearly full (i.e. α is very close to 1)

    Since the running time depends on α, we should decrease it by growing the array when α becomes too large

    Rule of thumb: if α increases beyond 0.5 or 0.75, grow the capacity

Running time of successful find?

- Chaining
  - What is the expected length of the longest chain? What is the average length of a chain?
  - Of course, we want our hash function to distribute keys well (if everything hashes to a constant number of buckets, lookup time would be O(n))
  - If you are lucky enough for the items to be uniformly distributed across buckets then the average length of chains would be $1/\alpha$
  - However, the birthday paradox from (see, Discrete Math) tells us that the probability of some collisions is high even if keys are drawn from uniform distribution
  - Therefore, α should still be kept sufficiently smaller than 1



Today's learning objectives

- Identify various data structures to implement a Set
- Calculate the memory usage of hashing data structures
- Execute Set methods for various hash set implementations, including when there are collisions
- Identify important properties of hash codes

Resources

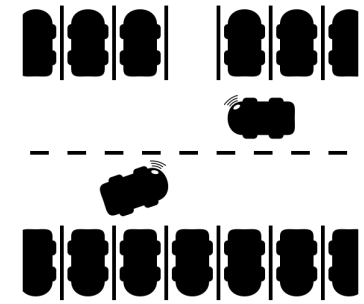
Visualizations of probing and chaining hash tables!

http://www.cs.usfca.edu/~galles/visualization/OpenHash.html

http://www.cs.usfca.edu/~galles/visualization/ClosedHash.html

http://www.cs.usfca.edu/~galles/visualization/ClosedHashBucket.html

Acknowledgements



Created by Jaime Carrion
from Noun Project

Cache diagrams – Ferry24Milan