

Programming Abstractions

CS106B

Instructor: Cynthia Bailey

Head TA: Yasmine Alonso

Today's Topics

pollev.com/cs106b

More ADTs!

- Map
 - › Code example: counting words in text
- Containers-within-containers
 - › Shallow copy vs. deep copy



Announcements

- No class Monday! MLK Jr Day holiday
 - › Recommended reading: MLK Jr's [Letter from a Birmingham Jail](#)
 - › Recommended watching: [MLK Jr's speech at Stanford \(!\)](#)

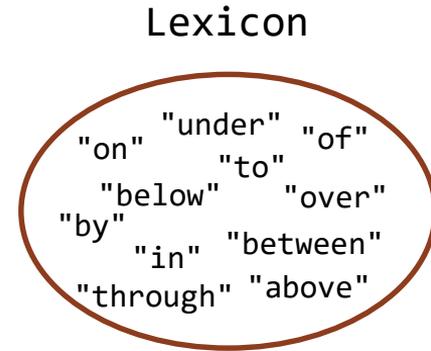
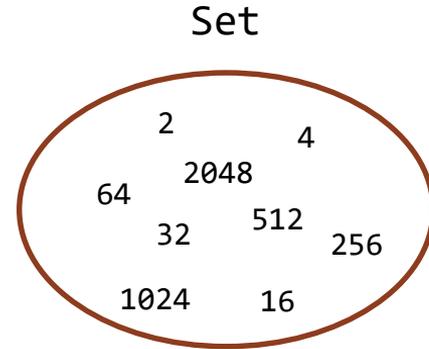
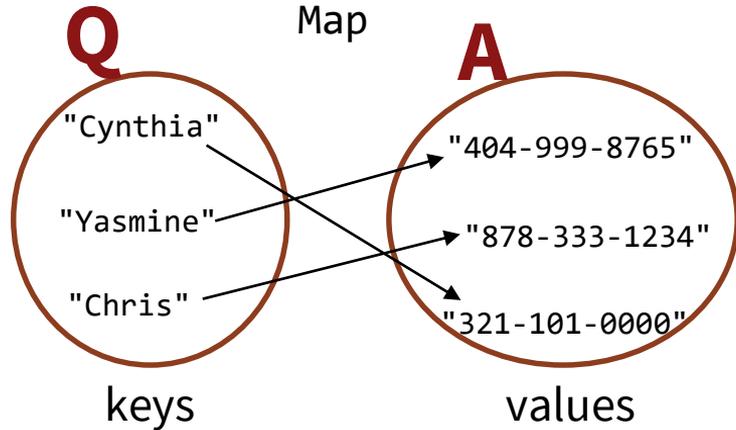
Maps

(not like the driving
directions kind of
maps though)



Associative containers

- Map
- Set
- Lexicon



Not as concerned with order but with association

- Map: associates **keys** with **values** (each could be any type)
- Set: associates **keys** with **membership** (in or out)
 - › Lexicon: a set of strings, *with special internal optimizations for that*

Stanford library Map *(selected member functions)*

```
void put(KeyType& key, ValueType& value);  
bool containsKey(KeyType& key);  
ValueType get(KeyType& key);  
ValueType operator [] (KeyType& key);
```

```
#include "map.h"
```

```
Map<string, string> phone;           // Map takes two(!) template parameters
```

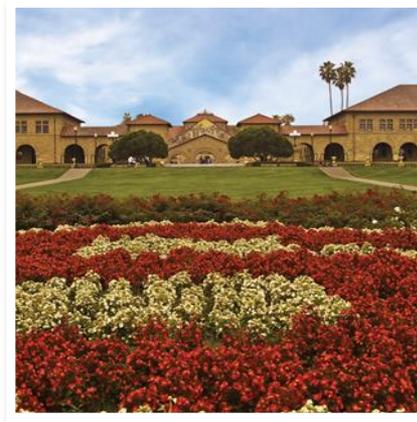
```
phone["Cynthia"] = "321-101-0000";  // two syntax options for adding new item  
phone.put("Yasmine", "878-333-1234");
```

```
if (phone.containsKey("Cynthia") && phone.containsKey("Yasmine")) {  
    cout << phone["Cynthia"] << endl;  // two syntax options for getting item  
    cout << phone.get("Yasmine") << endl;  
    cout << phone["Pres. Levin"] << endl; // what would this do??  
}
```



Map Code Example

Tabulating word
counts



Map programming exercise

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston). Then do two things with those counts:

- **Words that occur many times:**
 - › Print all words that appeared in the book at least 100 times, in alphabetical order
- **Query mode:**
 - › Give us a word and we report *how many times* that word appeared in the book (potentially call this function to query the counts many times)

Map programming exercise

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- The user gives us a word and we report *how many times* that word appeared in the book (potentially many such queries).

What would be a good design for this problem?

- A. `Map<int, string> wordCounts;`
- B. `Map<Vector<string>, Vector<int>> wordCounts;`
- C. `Map<Vector<int>, Vector<string>> wordCounts;`
- D. `Map<string, int> wordCounts;`
- E. `Map<string, Vector<int>> wordCounts;`
- F. Other/none/more

Go to
pollev.com/cs106b
to respond!



Stanford University

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

How can we record the count?

(In other words, what goes in the place marked “record count here” in the code at right?)

- A. `wordCounts[word] += word;`
- B. `wordCounts[word] += 1;`
- C. `wordCounts[word]++;`
- D. B and C are good, but you need to first detect new (never seen before) words so you can start at zero before you start adding +1
- E. Other/none/more

```
// We are given a vector that is just the
// the book, broken into pieces based on
// spaces between words. The type is:
// Vector<string> words;
```

```
Map<string, int> wordCounts;
for (string word : words) {
    // record count here
}
```



Go to
pollev.com/cs106b
to respond!

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- The user types a word and we report *how many times* that word appeared in the book (repeat in a loop until quit).

```
// userWord is a word the user typed into the console
cout << userWord << " appears " << wordCounts[userWord] << " times" << endl;
```

What happens if queryWord is not a word in the book?

- Will the program crash?
- What other issue(s) besides crash do you foresee?



Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- Report all words that appeared in the book at least 100 times, in alphabetical order

```
for (string word : wordCounts) {  
    if (wordCounts[word] >= FREQUENCY_THRESHOLD) {  
        cout << word << "\t" << wordCounts[word] << endl;  
    }  
}
```



Does this work for our alphabetical order requirement?

- Yes!
- Conveniently, Stanford library Map iterates over its keys in sorted order



Compound Containers

Containers within containers
within containers!
It's turtles all the way
down...



Can we add the number 4 to a Vector? Let's see...



```
Vector<int> numbers;  
numbers.add(1);  
numbers.add(2);  
numbers.add(3);  
Map<string, Vector<int>> mymap;  
mymap["abc"] = numbers;  
// Now we want to add 4 to the Vector inside the Map, how can we do it?
```

Go to
pollev.com/cs106b
to respond!

```
numbers.add(4);
```

```
mymap["abc"].add(4);
```

```
Vector<int> test = mymap["abc"];  
test.add(4);
```

Would any of these
three options would
work if inserted
here? Which one(s)?
Why or why not?

```
// GOAL: we want this to print 4 (indicating the add(4) worked)  
cout << "New size: " << mymap["abc"].size() << endl;
```

Can we add the number 4 to a Vector? Let's see...

You don't need to worry too much about the details of how the cases differ in terms of behind-the-scenes mechanism—I just wanted to flag it as a potential issue in case you accidentally encounter this in your code!

```
> mymap;
```

to the Vector inside the Map, how can we do it?

```
numbers.add(4);
```

```
mymap["abc"].add(4);
```

```
Vector<int> test = mymap["abc"];  
test.add(4);
```

Would any of these three options work if inserted here? Which one(s)? Why or why not?

```
// GOAL: we want this to print 4 (indicating the add(4) worked)  
cout << "New size: " << mymap["abc"].size() << endl;
```

How do we measure “faster” in Computer Science?

NOT AS SIMPLE AS YOU MIGHT
THINK...



**Recall our discussion of performance
with the Vector add vs. Insert...**

Your turn: Vector performance

- **Answer: (D) Something else! (about 50x)**
 - › In addition to analyzing the code and predicting number of operations, you should also time the code using our Stanford 106B test system.
 - › **Check the code bundle for class today for runnable version.**

Performance analysis
technique 1:
Benchmarking
(actually run it, and
time it)

```
void runInsert(int size)
{
    Vector<int> v;
    for (int i = 0; i < size; i++) {
        v.insert(0, i);
    }
}

void runAdd(int size)
{
    Vector<int> v;
    for (int i = 0; i < size; i++) {
        v.add(i);
    }
}
```

```
/* * * * * * Test Cases * * * * * */
PROVIDED_TEST("Timing comparison")
{
    int size = 500000;
    TIME_OPERATION(size, runInsert(size));
    TIME_OPERATION(size, runAdd(size));
}
```

```
SimpleTest VectorPerformance
Tests from PROVIDED_TEST

Correct (PROVIDED_TEST, vectortest.cpp:42) Timing comparison of add() at the end and insert() at the beginning
Line 44 TIME_OPERATION runInsert(size) (size = 500000) completed in 18.031 secs
Line 45 TIME_OPERATION runAdd(size) (size = 500000) completed in 0.030 secs

Tests from STUDENT_TEST

Correct (STUDENT_TEST, vectortest.cpp:48)

Tests from vectortest.cpp

Correct (PROVIDED_TEST, line 42) Timing comparison of add() at the end and insert() at the beginning
Line 44 TIME_OPERATION runInsert(size) (size = 500000) completed in 17.987 secs
Line 45 TIME_OPERATION runAdd(size) (size = 500000) completed in 0.029 secs

Correct (STUDENT_TEST, line 48)
```

Your turn: Vector performance

- **Answer: (D) Something else! (about 50x)**

- › Number of times a number is written in a box
 - OPTION 1:
 - First loop iteration: 1 write
 - Next loop iteration: 2 writes ... continued...
 - Formula for sum of numbers 1 to N = $(N * (N + 1)) / 2$
 - *(don't worry if you don't know this formula, we only expected a ballpark estimate)*
 - $100 * (100 + 1) / 2 = 10,100 / 2 = \mathbf{5,050}$
 - OPTION 2:
 - First loop iteration: 1 write
 - Next loop iteration: 1 write ... continued...
 - **100**

Performance analysis
technique 2:

Counting the precise
number of writes to
memory

Big-O: our primary performance analysis technique

- Big-O analysis in computer science is a way of counting the number of “steps” needed to complete a task
 - › Doesn’t really consider how “big” each step is
 - › Doesn’t consider how fast the computer’s CPU or other hardware components are
 - › Doesn’t involve any actual measurement of the time elapsed for any real code in any way
- But despite all that, really useful for making broad comparisons between different approaches

Efficiency as a virtue?

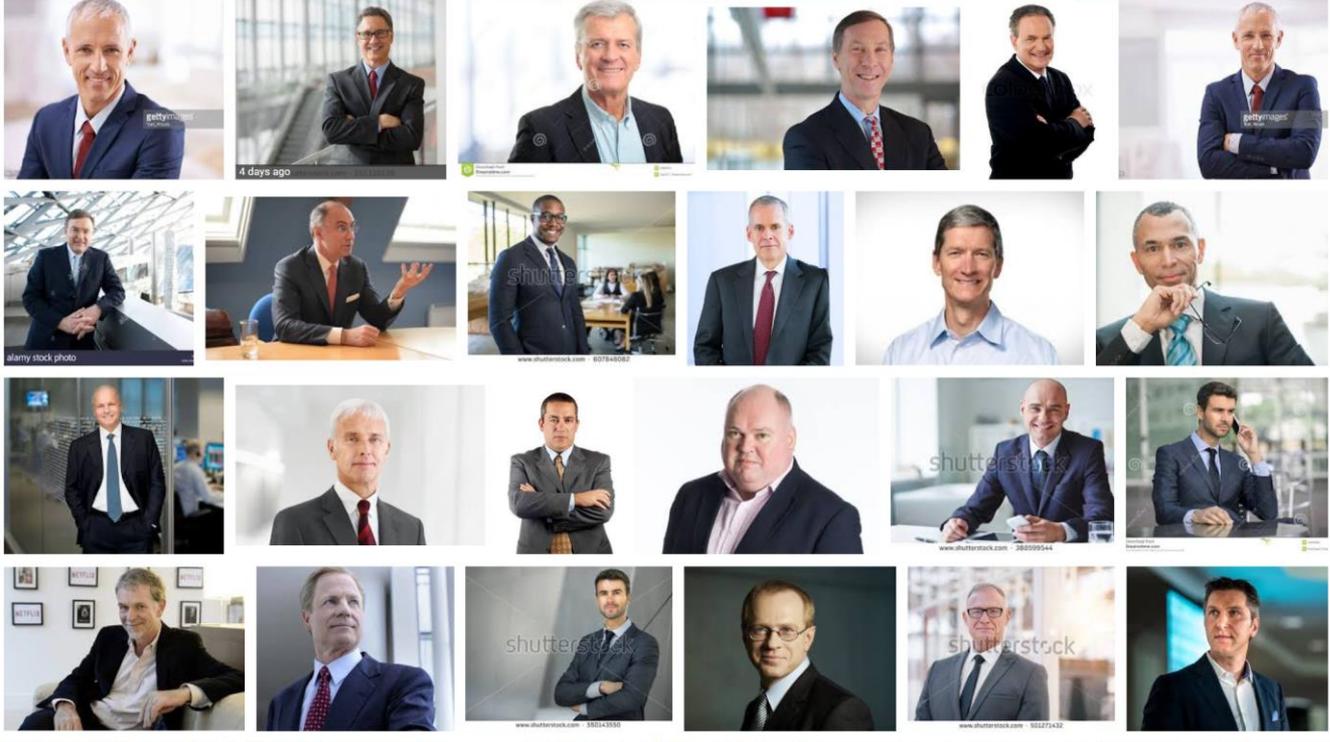
- In computer science, we tend to obsess about **efficiency**, but it's worth taking a step back and asking ourselves, is efficiency always a virtue?
 - › Racing to be first to the finish line, but with an answer that's wrong, isn't helpful!
 - › That might seem obvious, but it happens *all the time* in real tech products

Google image search

Google   

All **Images** News Videos Shopping More Settings Tools View saved SafeSearch

professional business linkedin shutterstock pfizer handsome netflix macy's reed hastings vanguard businessman ian c read david baazov 



4 days ago

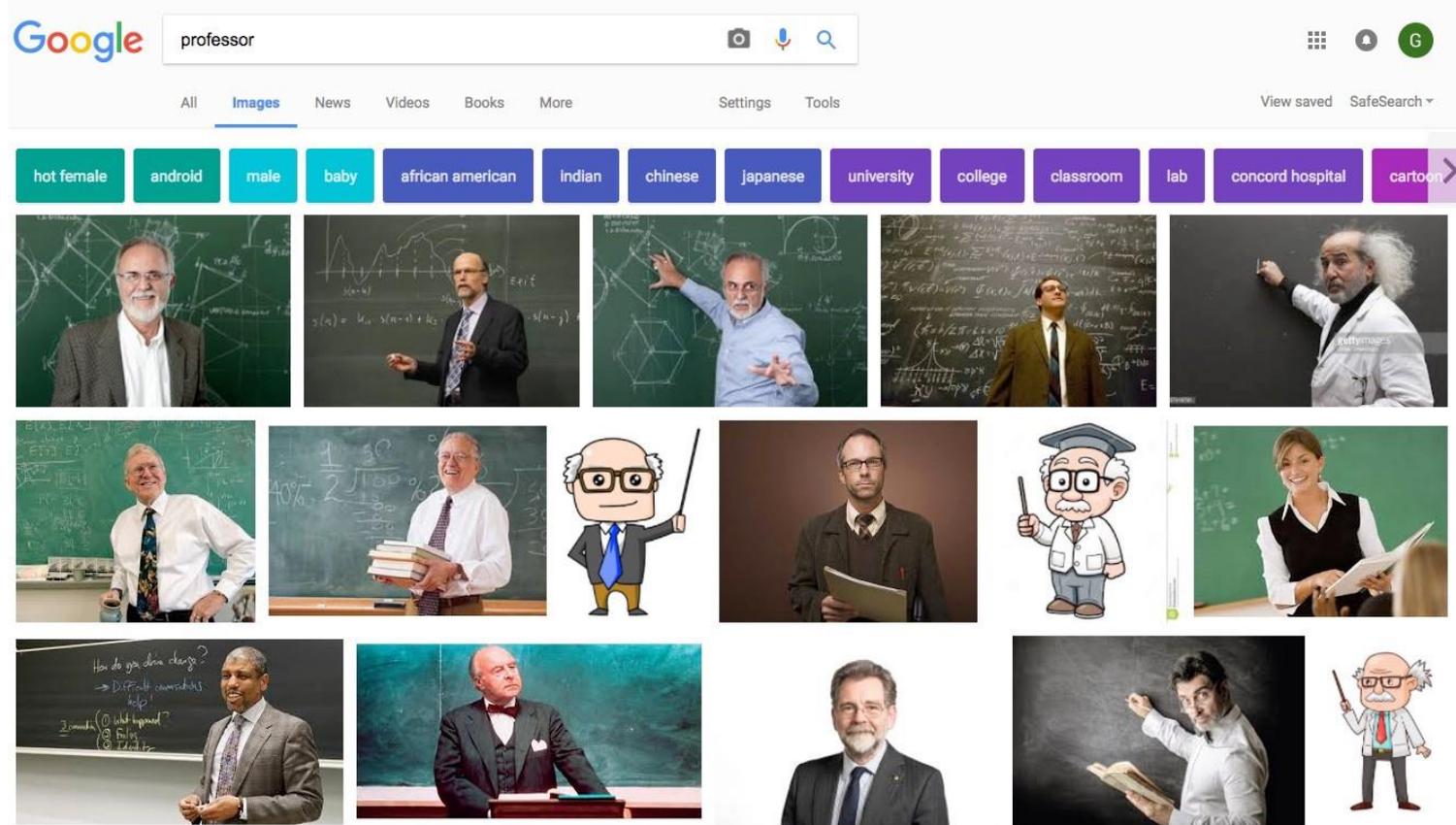
www.shutterstock.com 807840082

www.shutterstock.com 382995044

www.shutterstock.com 80243353

www.shutterstock.com 80272432

Another example...



The danger of a cheap solution: Twitter cropping

In the summer of 2020, Twitter users noticed something strange about Twitter's new photo cropping algorithm that is supposed to choose the most important/interesting part of the image

Given a too-tall image, it selects which part to show

It picked the Senator McConnell (the white man), not President Obama



The danger of a cheap solution: Twitter cropping

In the summer of 2020, Twitter users noticed something strange about Twitter's new photo cropping algorithm that is supposed to choose the most important/interesting part of the image

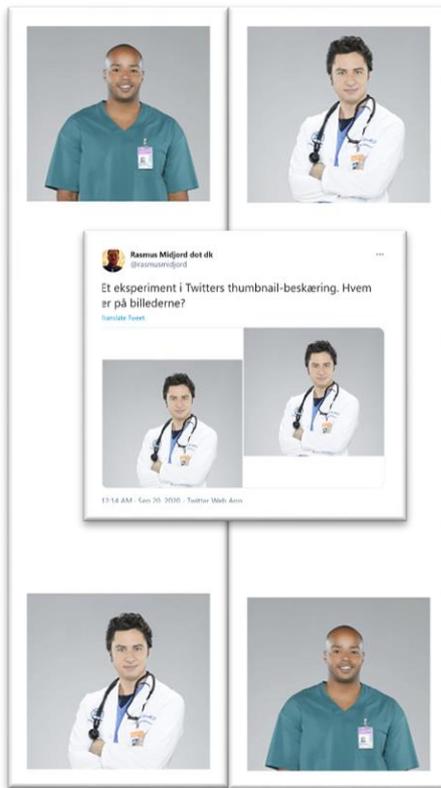
Given a too-tall image, it selects which part to show

It picked the Senator McConnell (the white man), not President Obama



Nope! It still picks McConnell when Obama is on top!





Efficiency as a virtue?

- In each of these cases, companies chose an algorithm that would reach the desired product outcome in a way that is the most *efficient*, for some business and/or computer science definition of efficient, but came up with answers that were “wrong” (problematic) in ways that are significant for society
- How can we balance cost (which is what efficiency is really about in capitalism) with correctness and justice for society?
- **Reflect on this in your Assignment 2!**