CS 600.226: Data Structures

Michael Schatz

Aug 30, 2018

Lecture 1: Introduction & Motivation

Welcome!

Course Webpage: https://github.com/schatzlab/datastructures2018
Course Discussions: https://piazza.com/jhu/fall2018/600226/home

Office Hours: Wednesday @ 2:45pm – 4pm, Malone 323
CA office hours throughout the week ☺

Programming Language: Java with Checkstyle and JUnit
Virtual Machine (Lubuntu) or CS acct.

Accounts for Majors (CS/CE) & Minors:

If you do not already have a personal CS departmental unix account, please complete an account request form ASAP. Check "Linux Undergrad" for account type. (Note - must be declared to be eligible.)

Accounts for Others:

We will need to make accounts. Do people need them?

CS Lab access:

Students must see Steve DiBlasio, with your J-card, in Malone G61A to get CS Lab access. The CS Lab is Malone 122 and that's where course TA/CAs will be available for help.

References and Resources

Primary Texts (Recommended, not required):

- (on-line interactive) OpenDSA, JHU version
- (print) Clifford A. Shaffer, Data Structures and Algorithm Analysis (Java Version) (Edition 3.2), available on-line and through Dover Publications.
- Peter Froehlich's Lecture notes posted to Piazza

Alternate Texts:

- Sedgewick & Wayne, Algorithms: JHU Library online edition
- Weiss, Data Structures and Algorithm Analysis in Java

Other Resources:

- Google ☺
- Code examples from Intro Programming in Java (600.107) - look in the sub-directories for examples of each topic.
- algoviz.org collection of visualizations for various data structures and algorithms
- Java API -- description of classes and methods

Grading and Help

Assessments:

- Weekly Assignments: 50% Due at 11:59pm ~one week later
- Midterm: 20% In class (~Friday Oct 12)
- Final Exam: 30% During exam week (Date TBD)

- In-class: Not graded, but there to help you!

Policies:

- Percentile scores assigned relative to the highest points awarded
- Fixed cutoffs for A+(>97); A(>93); A- (>90); B+ (>87); B (>83); B- (>80); etc
- Automatic testing and grading of coding assignments using gradescope

- ***Grace period:*** 10% penalty for up to 1 hour late
- ***Late Days:*** Five (5) chances to extend the deadline by 24 hours without any penalty

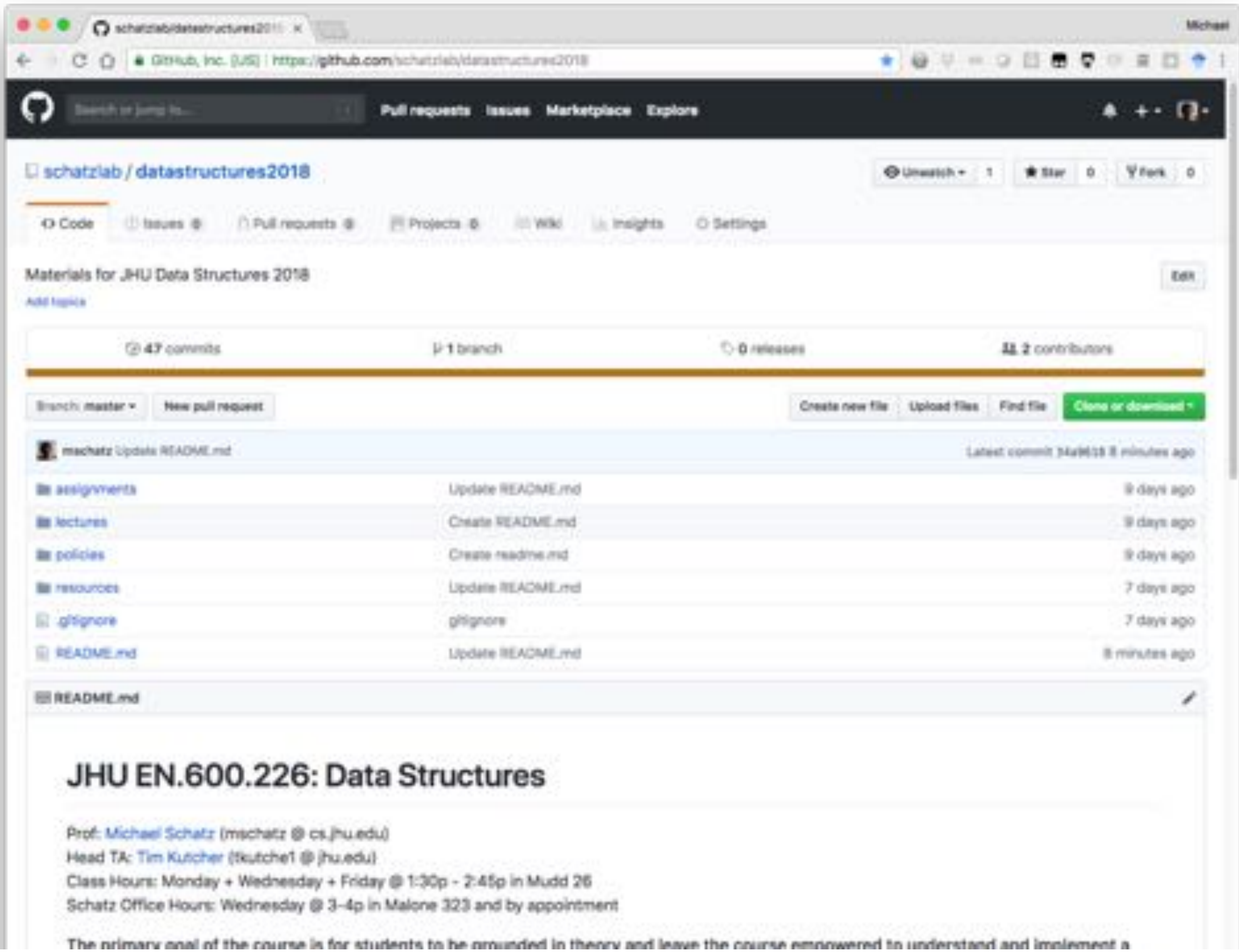
WARNING: If you submit >1 hour late and you don't have a late day left, then you will receive 0 points

Details:

https://github.com/schatzlab/datastructures2018/tree/master/policies

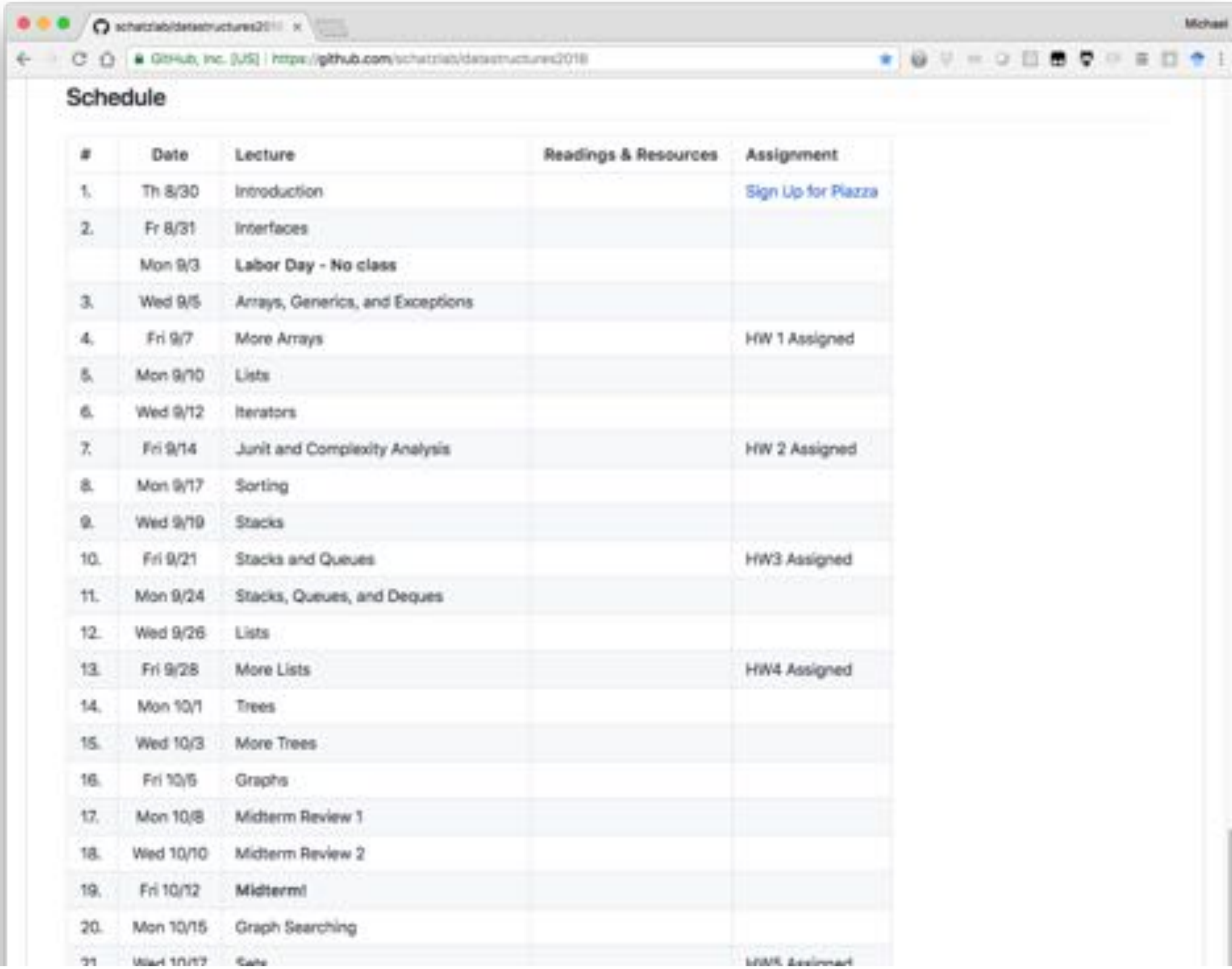
Course Webpage

https://github.com/schatzlab/datastructures2018



Course Webpage

https://github.com/schatzlab/datastructures2018

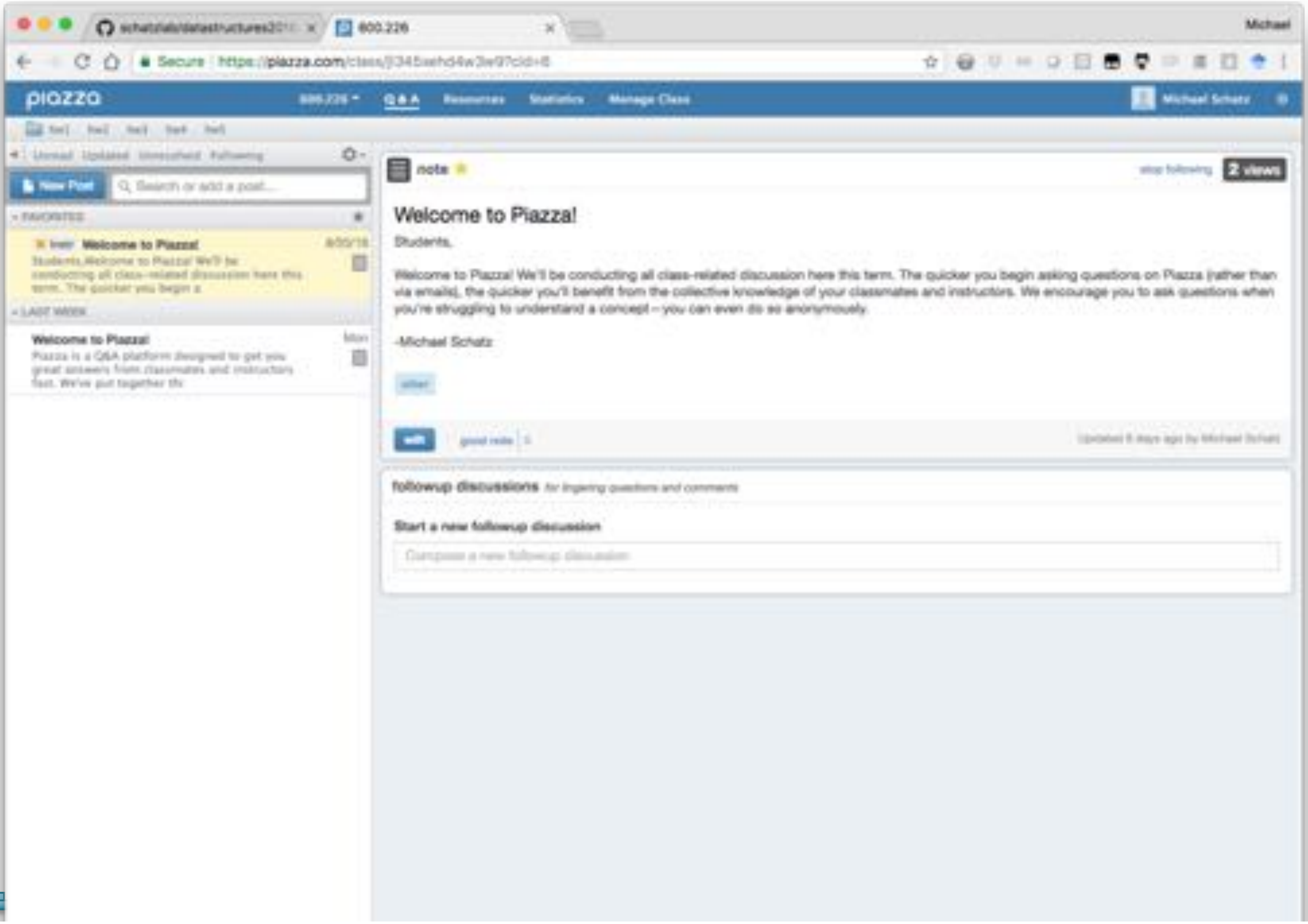


Schedule

#	Date	Lecture	Readings & Resources	Assignment
1.	Th 8/30	Introduction		Sign Up for Piazza
2.	Fr 8/31	Interfaces		
	Mon 9/3	**Labor Day - No class**		
3.	Wed 9/5	Arrays, Generics, and Exceptions		
4.	Fri 9/7	More Arrays		HW 1 Assigned
5.	Mon 9/10	Lists		
6.	Wed 9/12	Iterators		
7.	Fri 9/14	Junit and Complexity Analysis		HW 2 Assigned
8.	Mon 9/17	Sorting		
9.	Wed 9/19	Stacks		
10.	Fri 9/21	Stacks and Queues		HW3 Assigned
11.	Mon 9/24	Stacks, Queues, and Deques		
12.	Wed 9/26	Lists		
13.	Fri 9/28	More Lists		HW4 Assigned
14.	Mon 10/1	Trees		
15.	Wed 10/3	More Trees		
16.	Fri 10/5	Graphs		
17.	Mon 10/8	Midterm Review 1		
18.	Wed 10/10	Midterm Review 2		
19.	Fri 10/12	**Midterm!**		
20.	Mon 10/15	Graph Searching		
21.	Wed 10/17	Sets		HW5 Assigned

Piazza

https://piazza.com/jhu/fall2018/600226/home



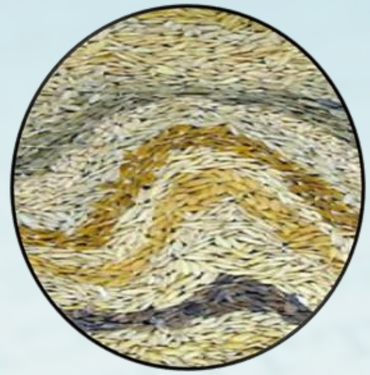
A Little About Me

Schatzlab Overview

Human Genetics

Role of mutations in disease

Nattestad et al. (2018)
Feigin *et al.* (2017)



Agricultural Genomics

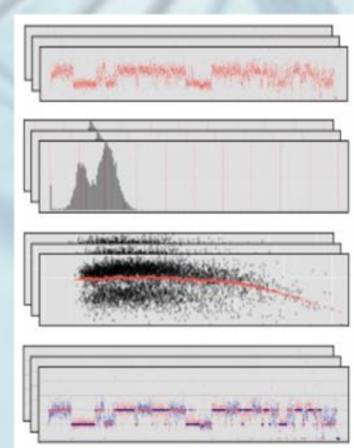
Genomes & Transcriptomes

Lemmon *et al.* (2016)
Ming *et al.* (2015)

Algorithmics & Systems Research

Ultra-large scale biocomputing

Stevens *et al.* (2015)
Marcus *et al.* (2014)



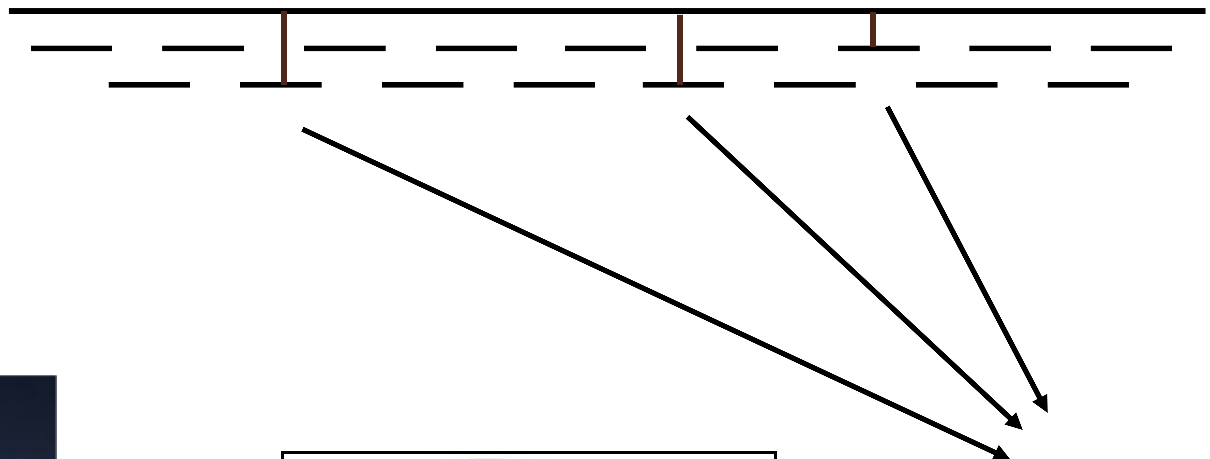
Single Cell & Single Molecule

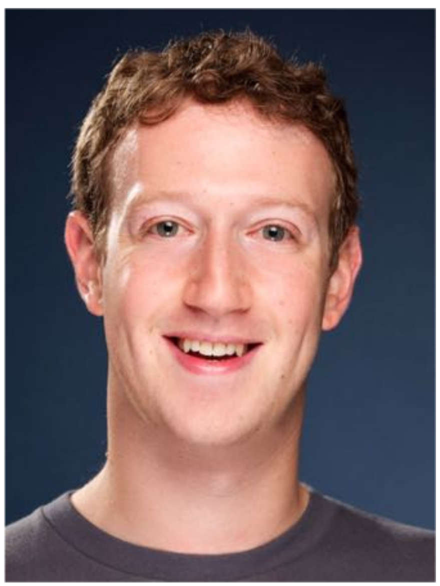
CNVs, SVs, & Cell Phylogenetics

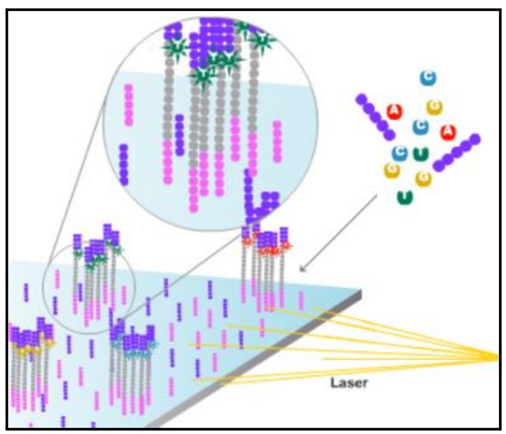
Sedlazeck *et al.* (2018)
Garvin *et al.* (2015)

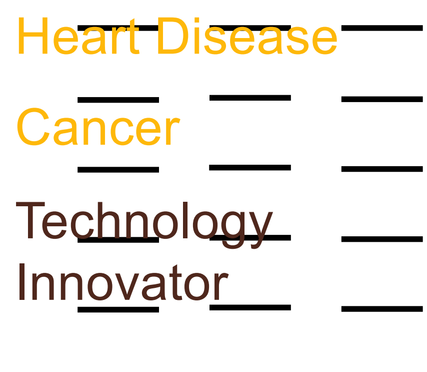
Personal Genomics

How does your genome compare to the reference?

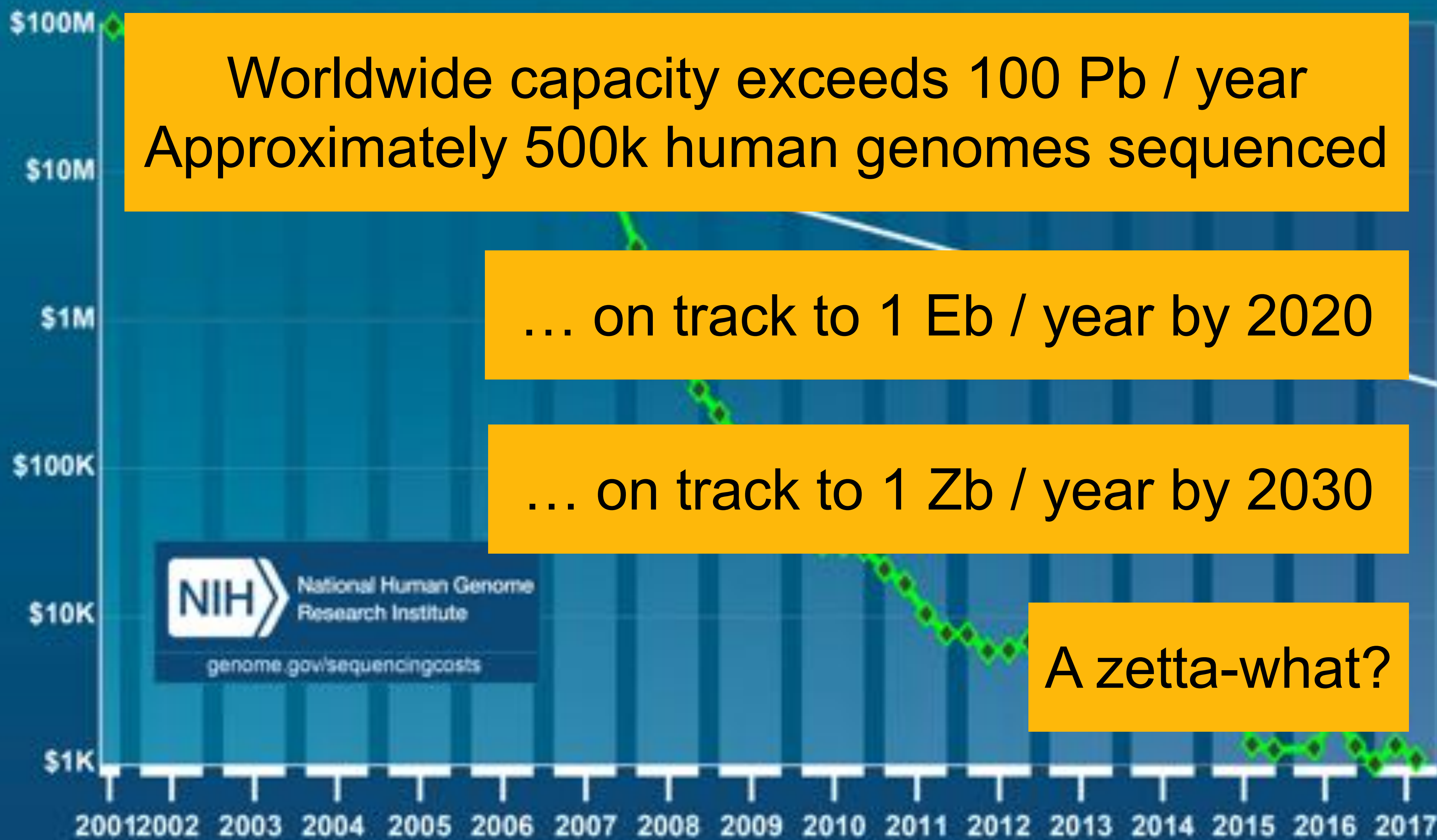








Cost per Genome



Worldwide capacity exceeds 100 Pb / year
Approximately 500k human genomes sequenced

… on track to 1 Eb / year by 2020

… on track to 1 Zb / year by 2030

A zetta-what?

http://www.genome.gov/sequencingcosts/

How much is a zettabyte?

Unit	Size	~2^x
Byte	1	2^0
Kilobyte	1,000	2^{10}
Megabyte	1,000,000	2^{20}
Gigabyte	1,000,000,000	2^{30}
Terabyte	1,000,000,000,000	2^{40}
Petabyte	1,000,000,000,000,000	2^{50}
Exabyte	1,000,000,000,000,000,000	2^{60}
Zettabyte	1,000,000,000,000,000,000,000	2^{70}

How much is a zettabyte?

100 GB / Genome
4.7GB / DVD
~20 DVDs / Genome

X

10,000,000,000 Genomes

=

1ZB Data
200,000,000,000 DVDs

150,000 miles of DVDs
~ ½ distance to moon

YouTube

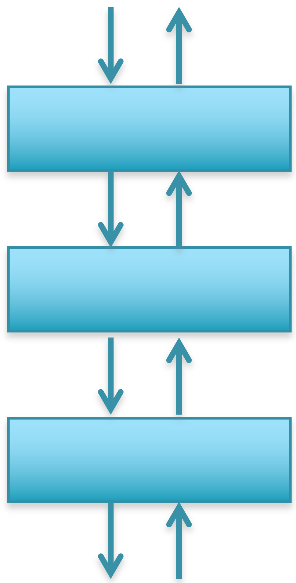
Both currently ~100PB
And growing exponentially

Science
11 February 2011 | $10
scientific
data
research
information
climate
science
new
analysis
visualization
researchers
access
many
human
knowledge
example
sharing
scientists
brain
challenges
model
need
use
cell
AAAS

LinkedIn
foursquare
LIVEJOURNAL
vimeo
YouTube
facebook
twitter
flickr
tumblr.
Blogger
Like
yelp
Forecasting the race for the House
5 in 7
Chance Democrats win control (73.1%)
2 in 7
Chance Republicans keep control (26.9%)
+33 Democratic seats
10% chance Democrats gain more than 55 seats
80% chance Democrats gain 14 to 55 seats
10% chance Democrats gain fewer than 14 seats
information
climate
science
sharing
model
brain
challenges
analysis
visualization
researchers
health
community
many
new
Google Books Ngram Viewer

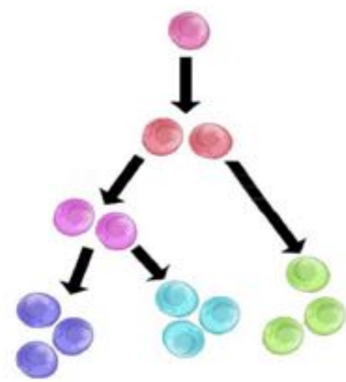
Data Structures

Lists



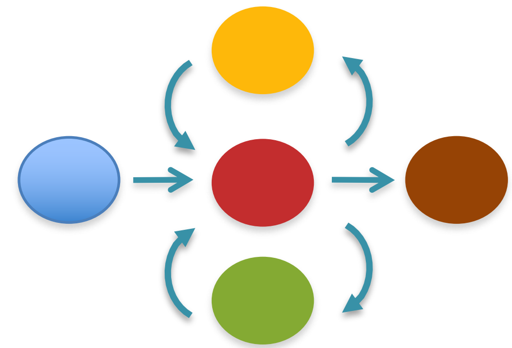
- Single/Double
- Stacks/Queues/Deques
- Skip

Trees



- Binary, AVL Trees
- Heaps
- Self Balancing

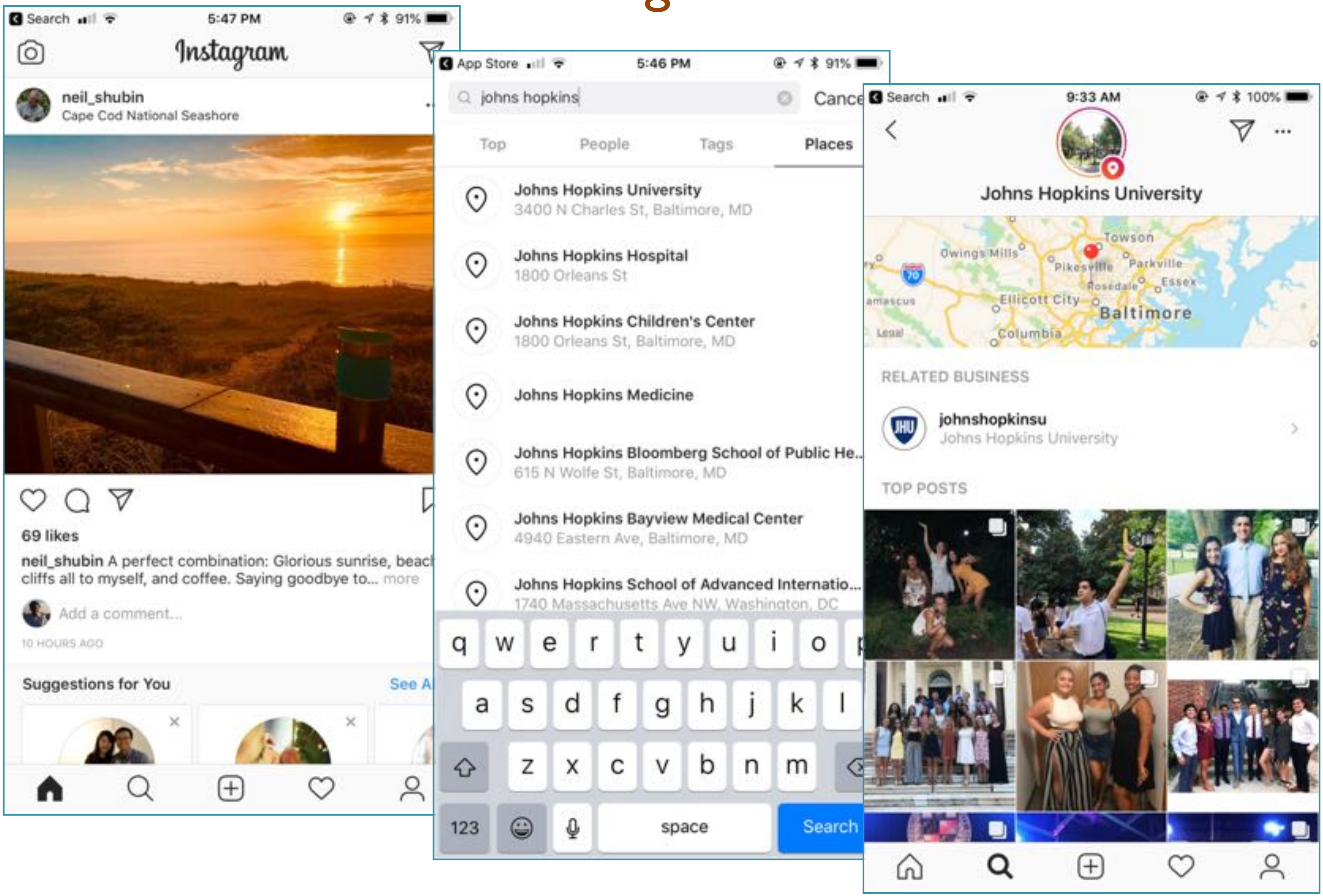
Graphs



- Graph Representations
- Traversing
- Union Find

Building, searching, traversing, analyzing
Make you big-data superheros ☺

Instagram



Search
5:47 PM
91%
Instagram
neil_shubin
Cape Cod National Seashore
69 likes
neil_shubin A perfect combination: Glorious sunrise, beac
cliffs all to myself, and coffee. Saying goodbye to... more
Add a comment...
10 HOURS AGO
Suggestions for You
See A
App Store
5:46 PM
91%
johns hopkins
Cance
Top
People
Tags
Places
Johns Hopkins University
3400 N Charles St, Baltimore, MD
Johns Hopkins Hospital
1800 Orleans St
Johns Hopkins Children's Center
1800 Orleans St, Baltimore, MD
Johns Hopkins Medicine
Johns Hopkins Bloomberg School of Public He...
615 N Wolfe St, Baltimore, MD
Johns Hopkins Bayview Medical Center
4940 Eastern Ave, Baltimore, MD
Johns Hopkins School of Advanced Internatio...
1740 Massachusetts Ave NW, Washington, DC
q w e r t y u i o
a s d f g h j k l
z x c v b n m
123
space
Search
Search
9:33 AM
100%
Johns Hopkins University
Towson
Owings Mills
Pikesville
Parkville
Rosedale
Essex
Ellicott City
Baltimore
Columbia
Legal
RELATED BUSINESS
johnshopkinsu
Johns Hopkins University
TOP POSTS

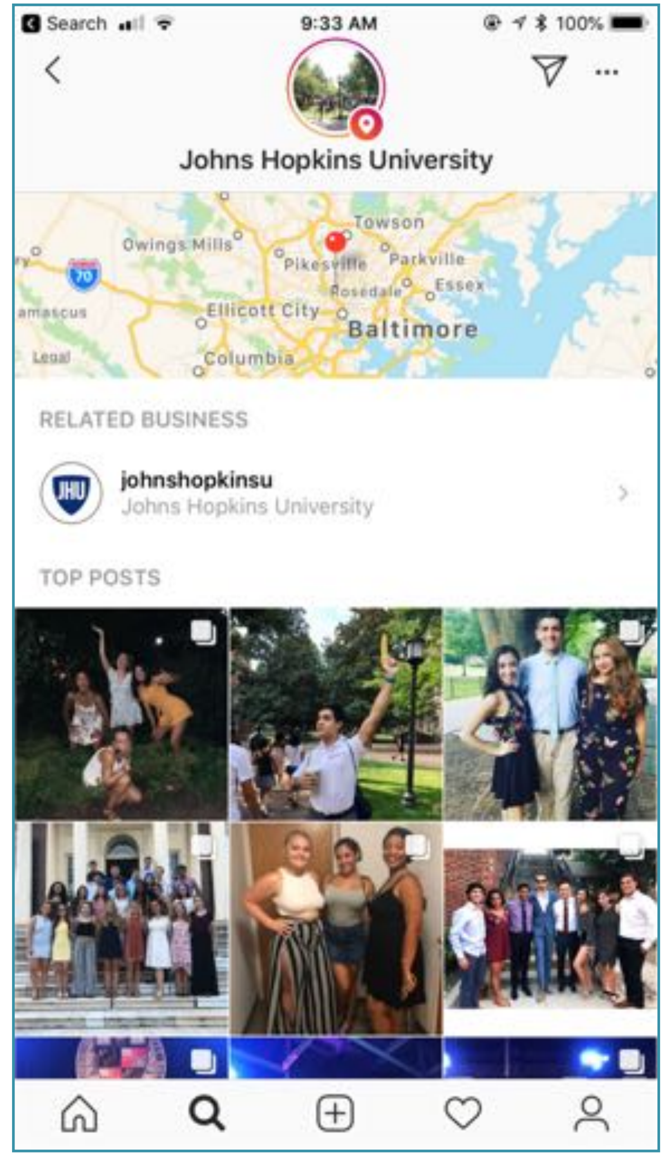
Data Structures of Instagram

Incredibly popular app:

~800M active users

>20B photos, >60M per day!

https://www.quora.com/How-many-photos-are-being-uploaded-on-Instagram-daily



How to find all photos near a given site?

Modern clock speed: 1 instruction / nanosec

Practical processing speed: 1000 photos / sec

1M seconds = ~11.5 days

20B photos / 1000 photos / s = 20M sec

= ~230 days

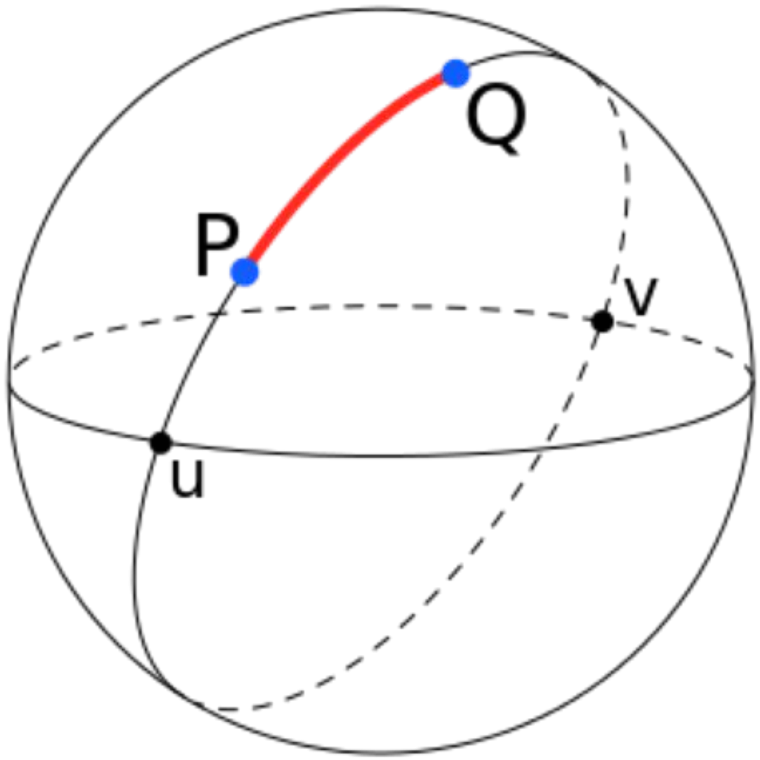
What if all users search at the same time?

230 days * 800M users = 184B days

= ~500M years

How can we make it go faster?

Data Structures of Instagram



$$\Delta\sigma = \arctan \frac{\sqrt{(\cos\phi_2 \cdot \sin(\Delta\lambda))^2 + (\cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos(\Delta\lambda))^2}}{\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\Delta\lambda)}.$$

https://en.wikipedia.org/wiki/Great-circle_distance

Inside Instagram

```
Search: JHU
Where: 39.32N 76.62W

Photo  #1
Where: 37.77N 122.41W (SFO)
URL: instagram.com/p/1

Photo  #2
Where: 20.63N 76.77W (Cuba)
URL: instagram.com/p/2
```

…

```
Photo  #3526224
Where: 39.32N 76.63W (JHU!)
URL: instagram.com/p/3526224
```

Show me the photos!

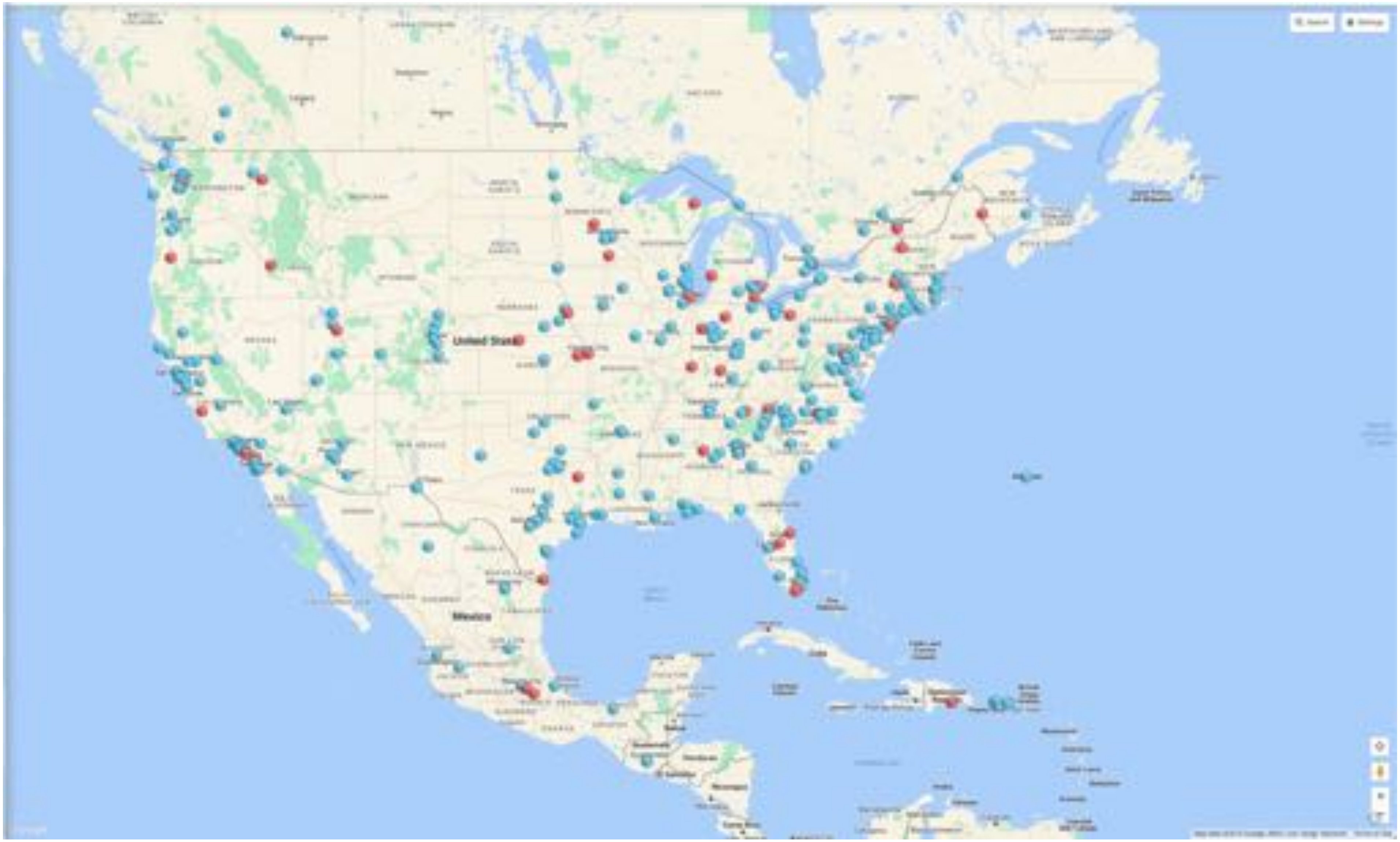
Linear Search (aka Brute force): try all 20B photos

```
#1: 37.77N 122.41W: No
#2: 20.63N 76.77W:  No
#3: 21.30N 157.85W: No
...
#3,526,224 39.32N 76.63W: Yes!
...
#19,999,999,999 48.85N 2.34E No
#20,000,000,000 35.65N 139.83E No
```

If you get really lucky you might find a few nearby photos quickly that you can return first

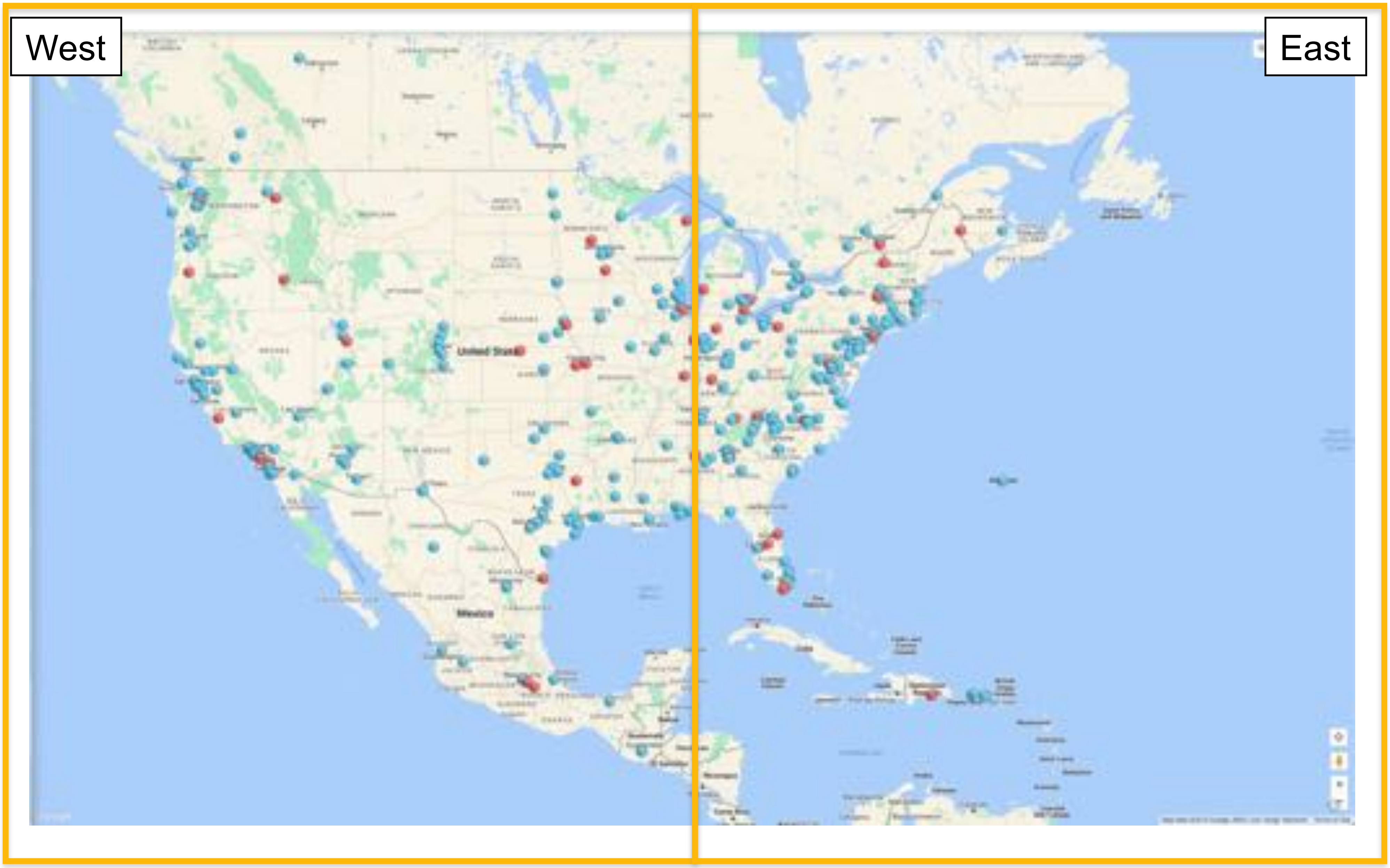
What happens if there are no photos at the search site?

Show me the photos!



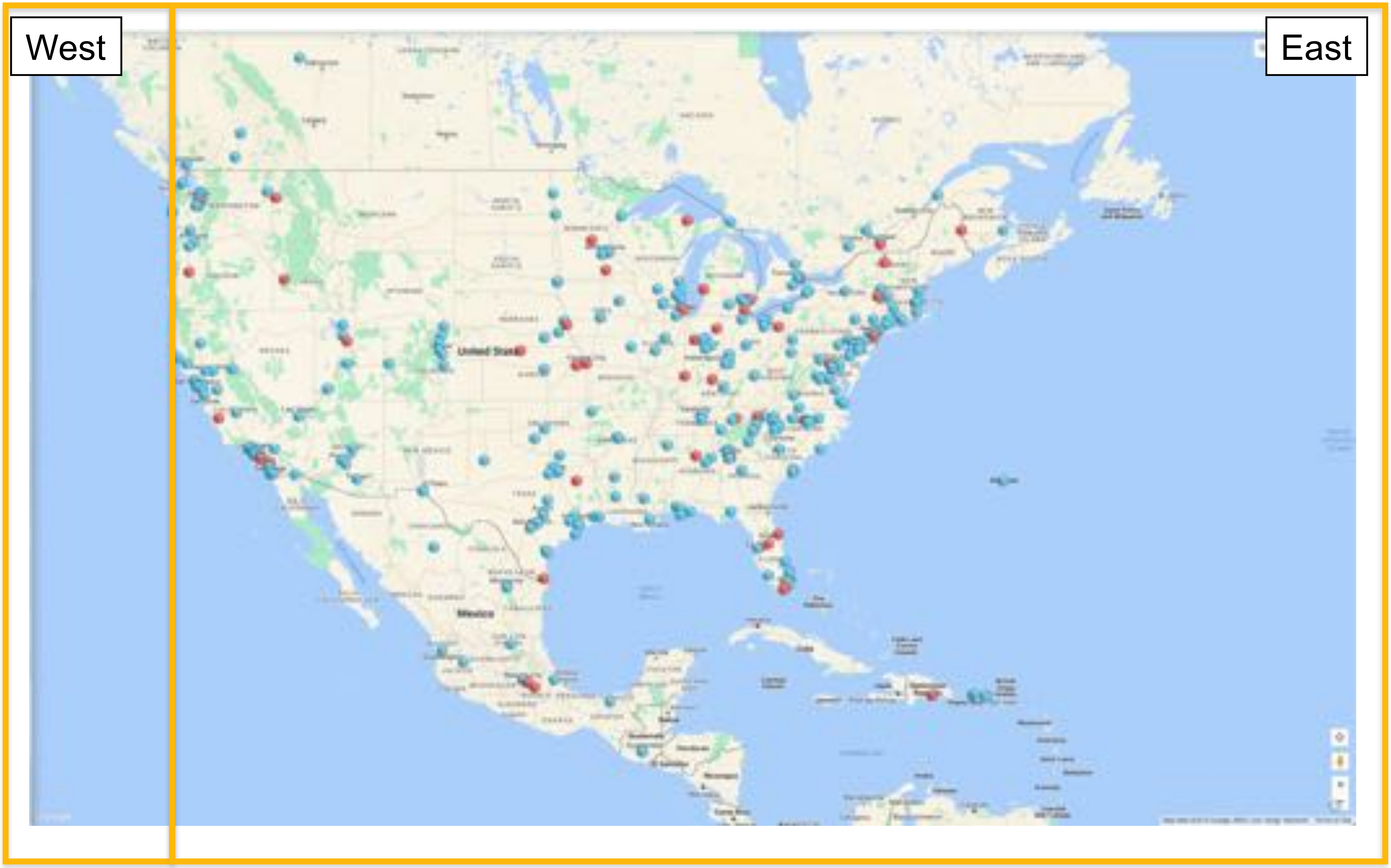
What can you do to speed up the search?
Note: The computer can only "see" one photo at a time

Show me the photos!



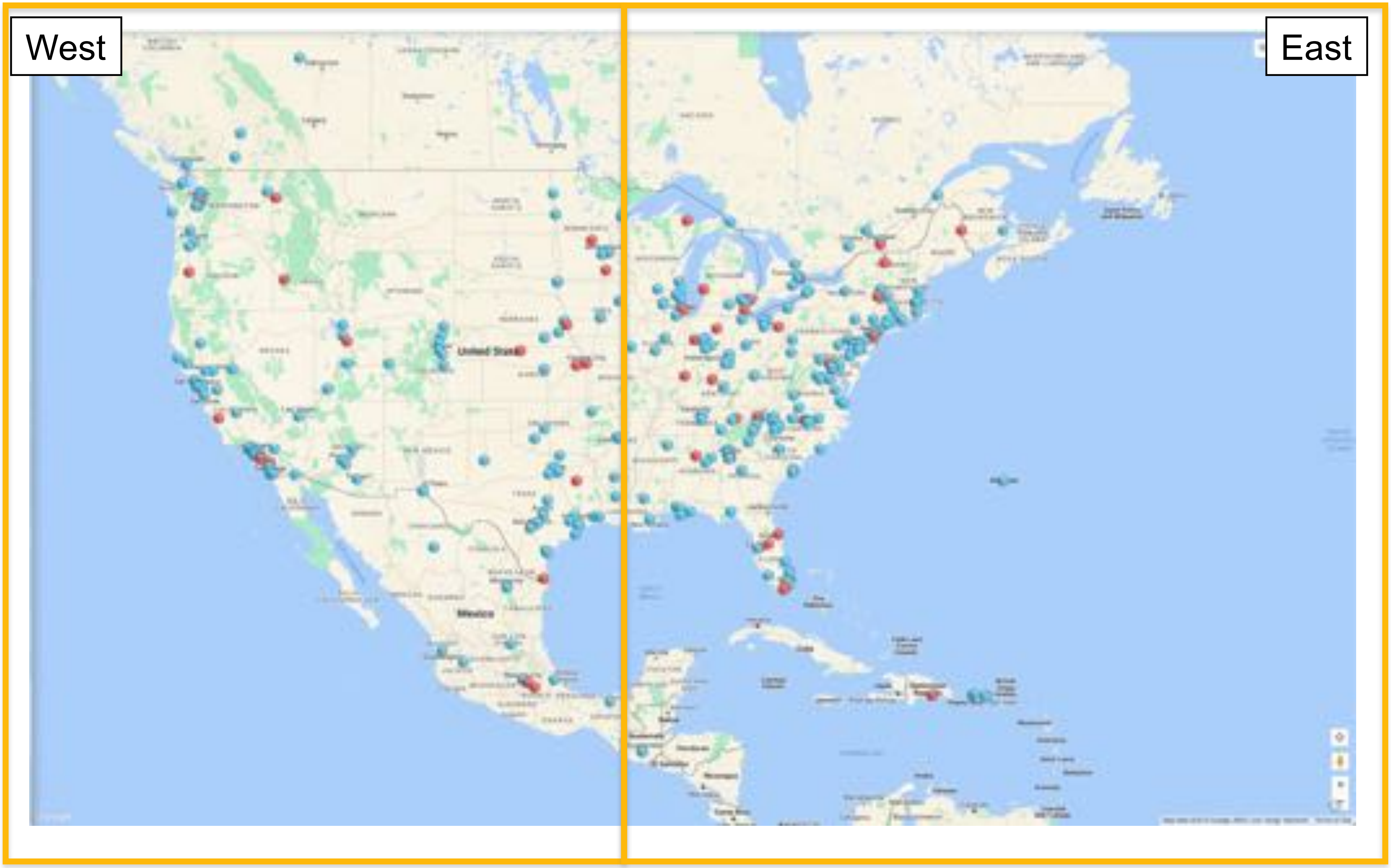
Partition the data into 2 lists, each search takes half as long!

Show me the photos!



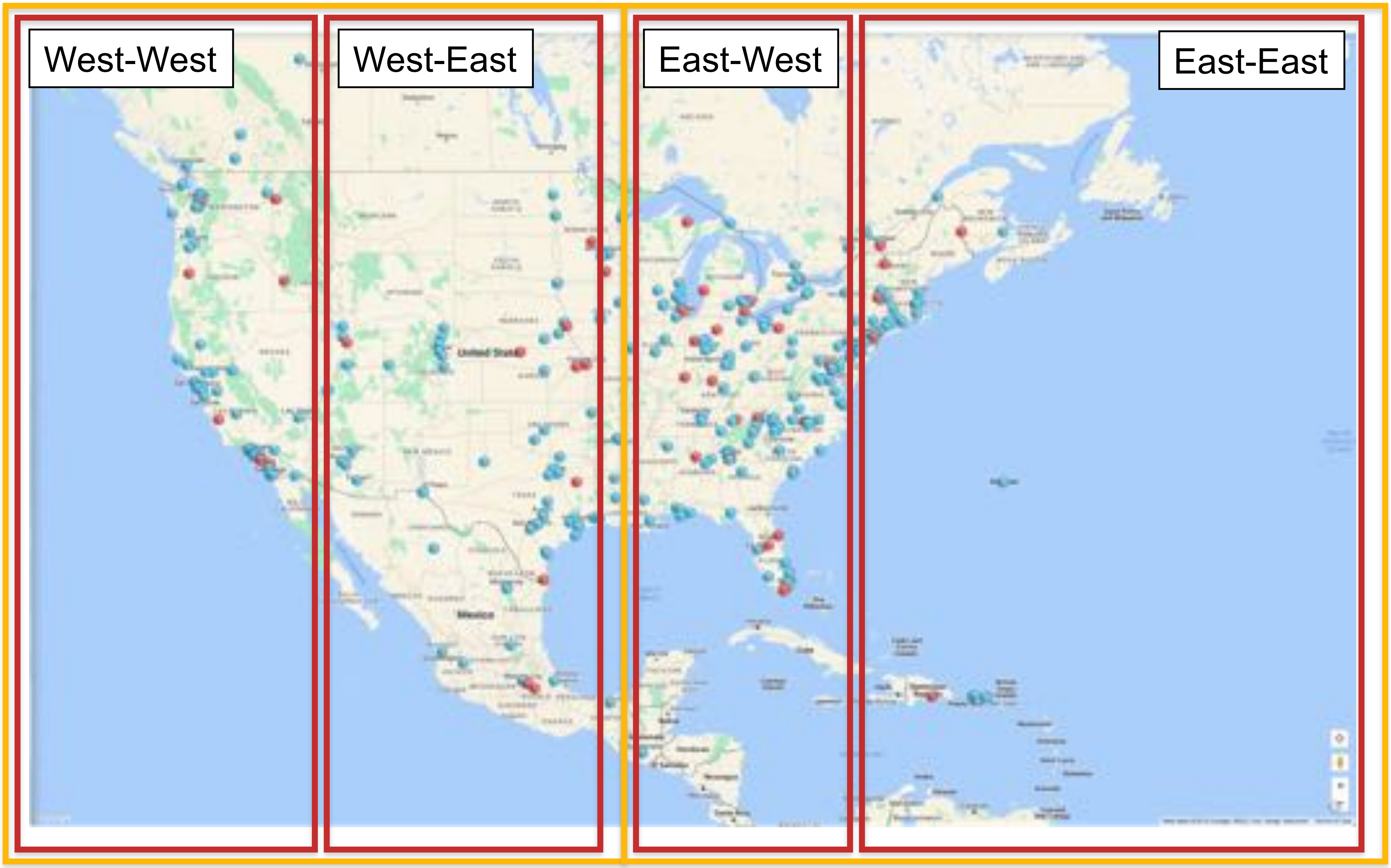
Why is this a bad split? What would be the perfect split?

Show me the photos!



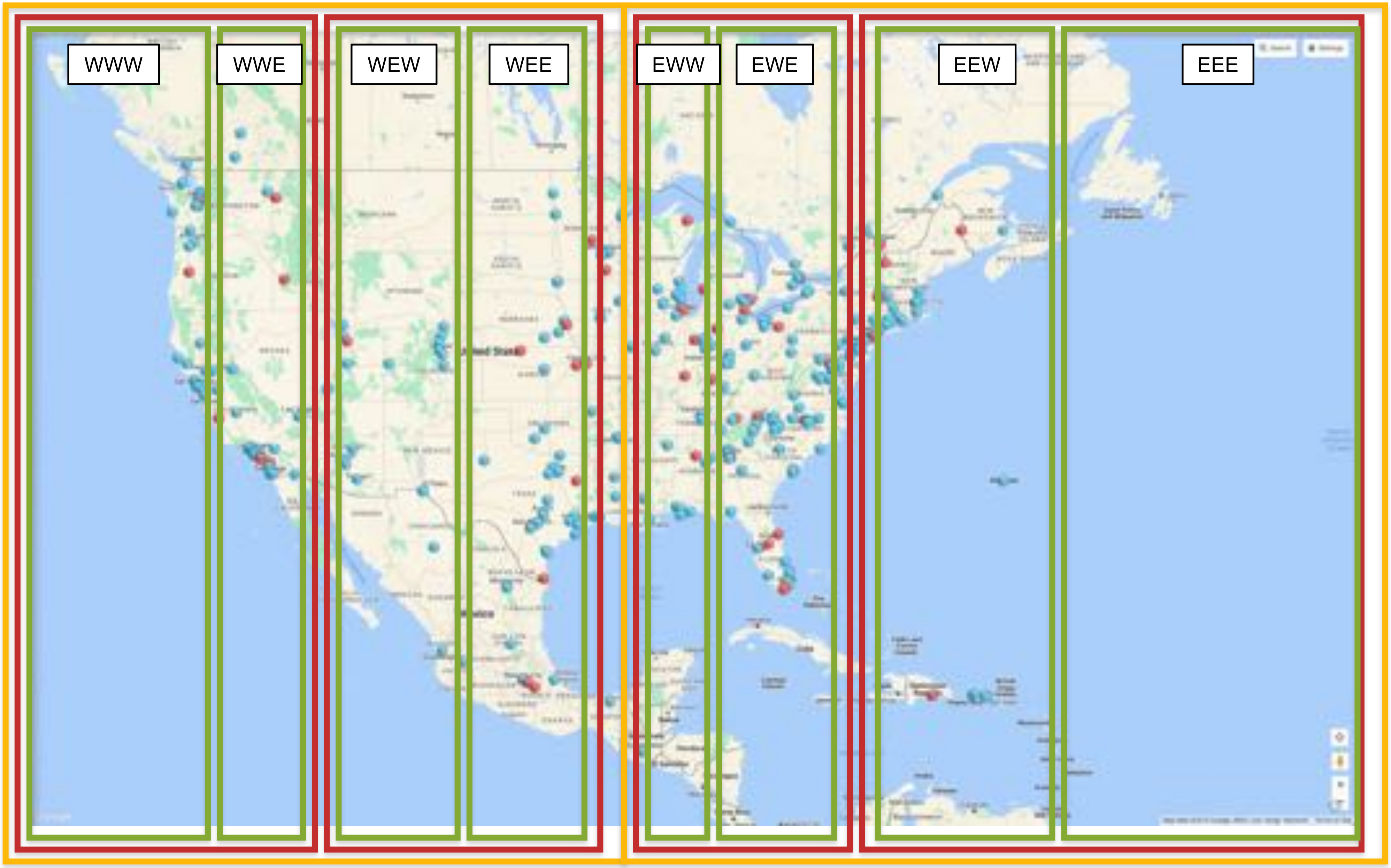
Ideal split will be exactly 50/50 (median east-west coordinate of sites)

Show me the photos!



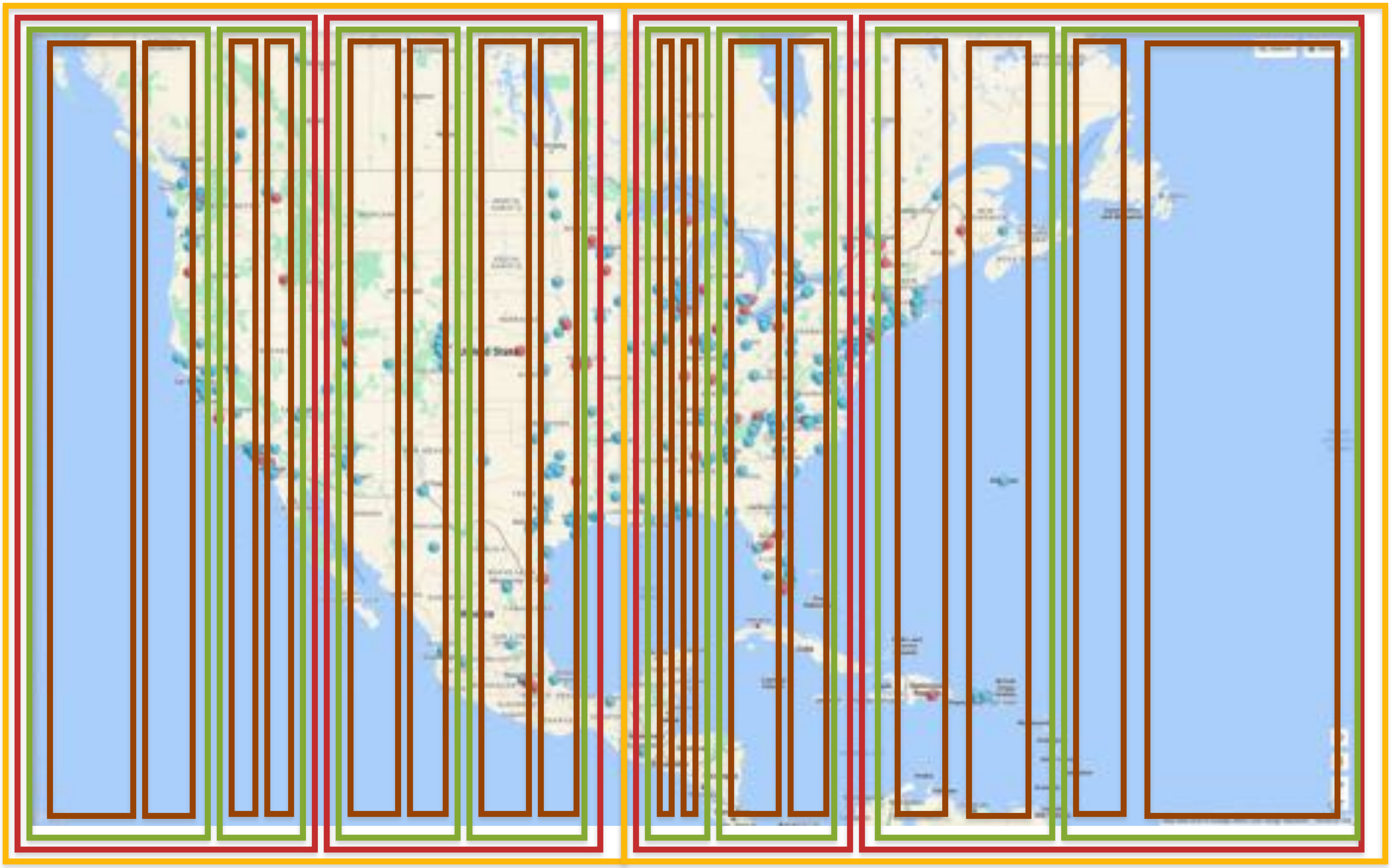
Partition again! Each sublist has N/4 elements!

Show me the photos!



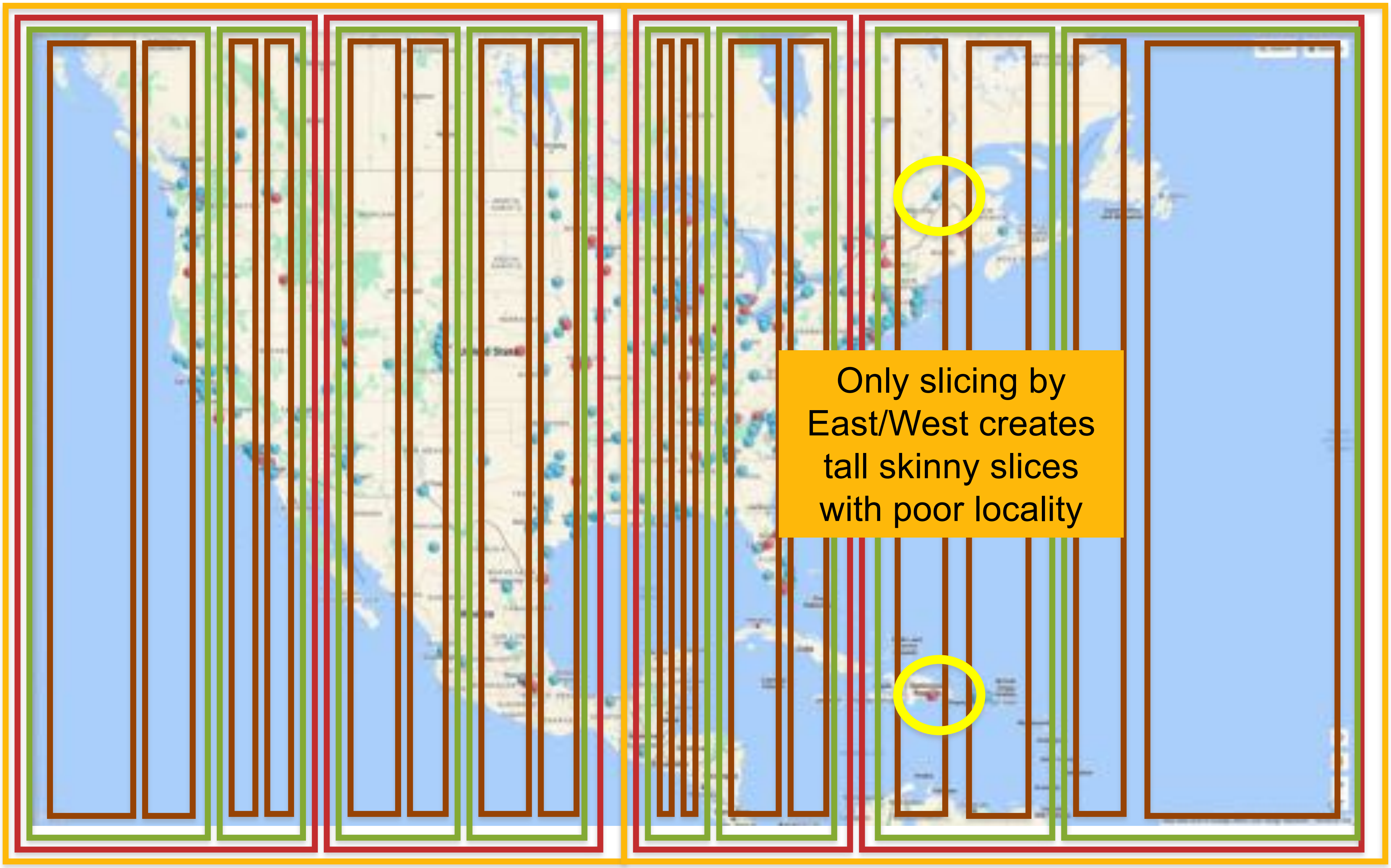
Partition again! Each sublist has N/8 elements!

Show me the photos!



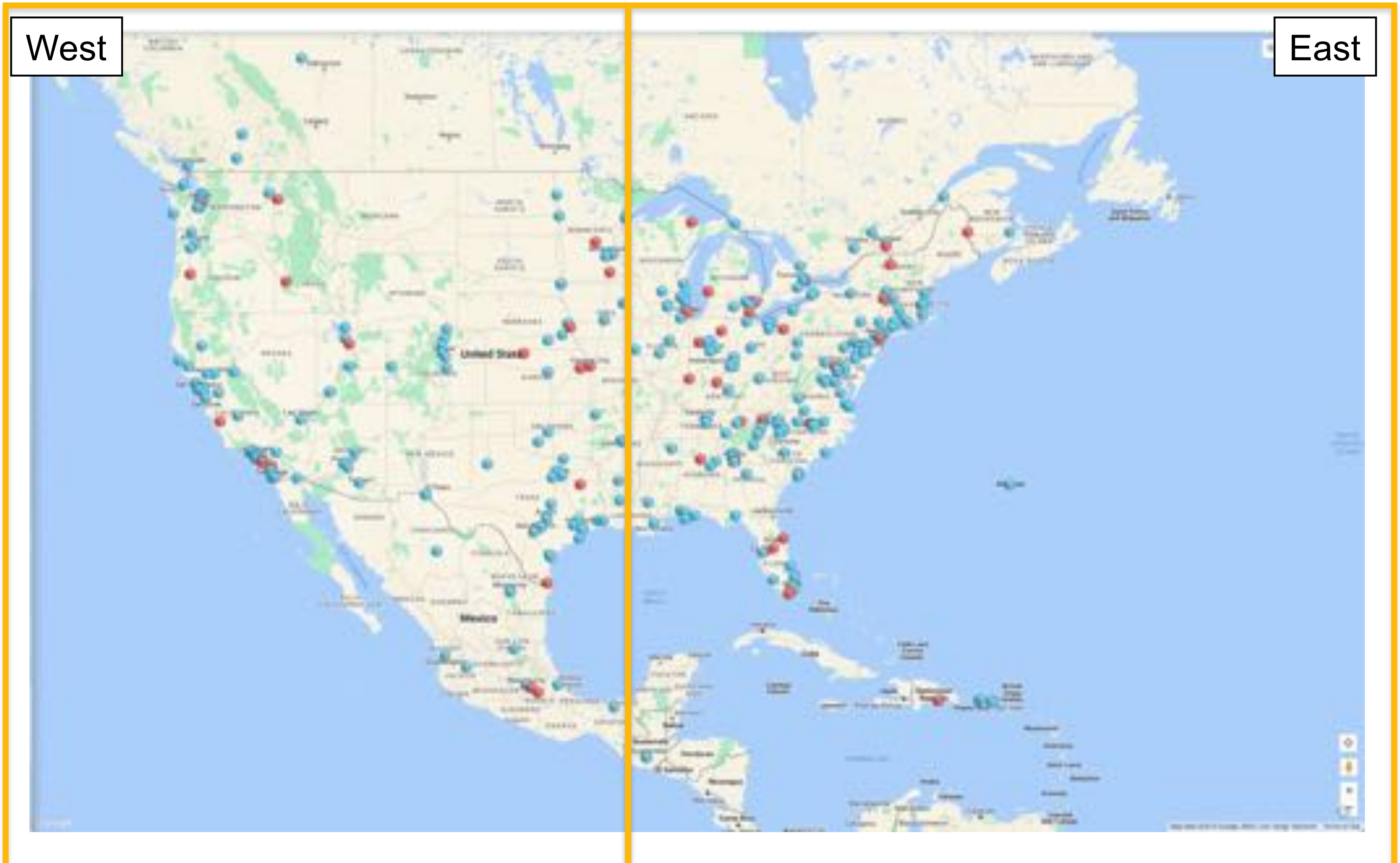
Partition again! Each sublist has N/16 elements!

Show me the photos!



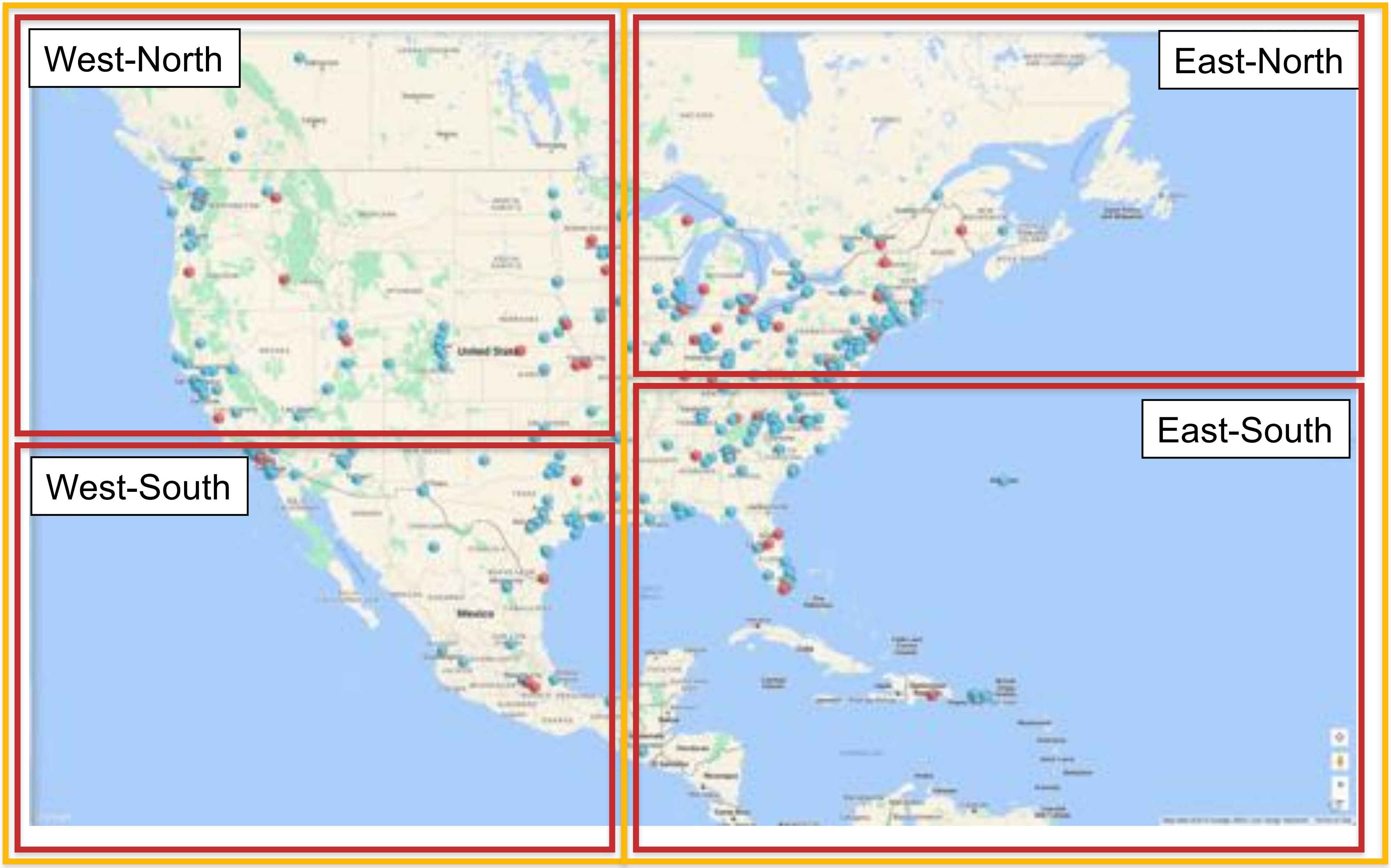
Partition again! Each sublist has N/16 elements!

Show me the photos!



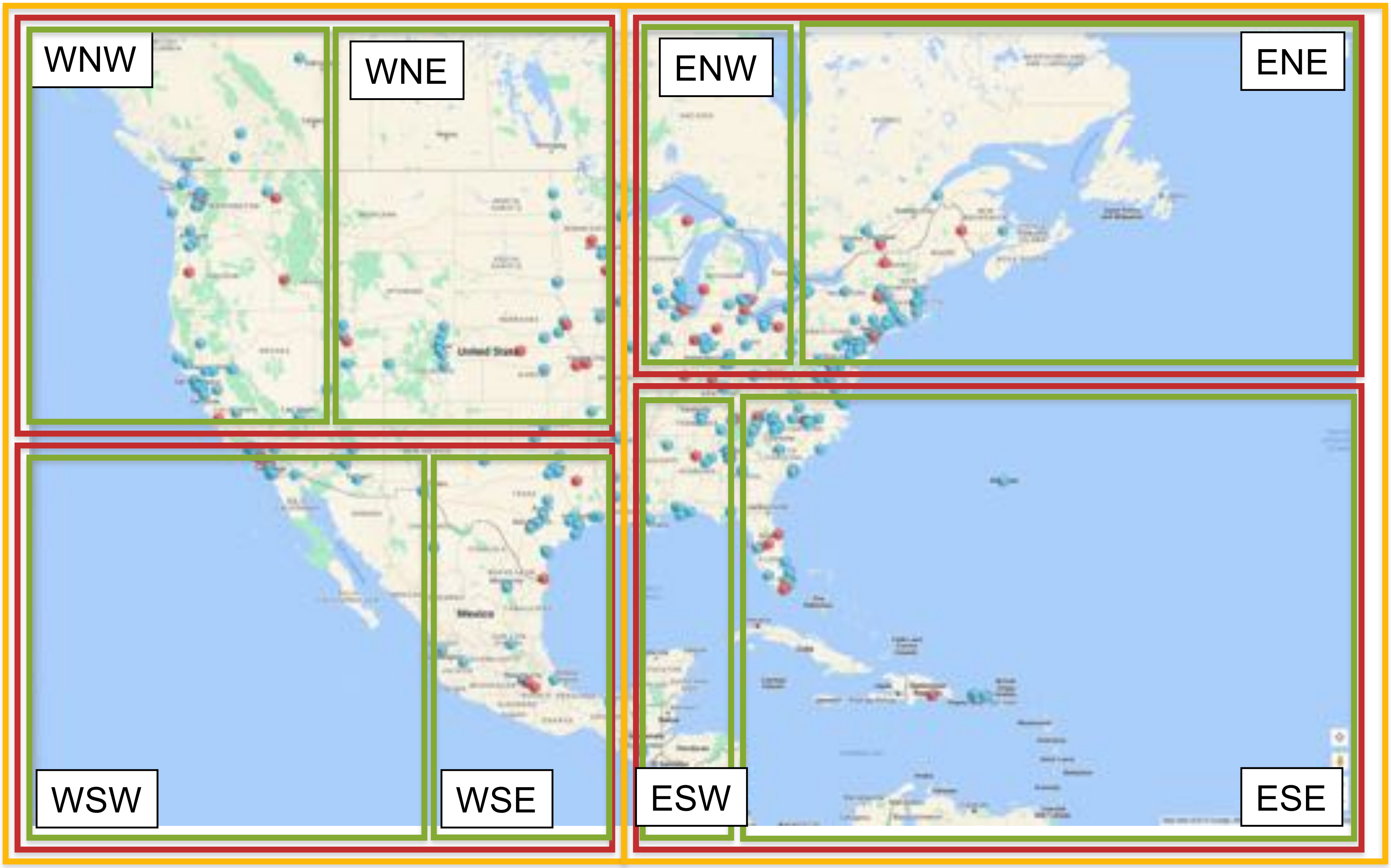
Ideal split will be exactly 50/50 (median east-west coordinate)

Show me the photos!



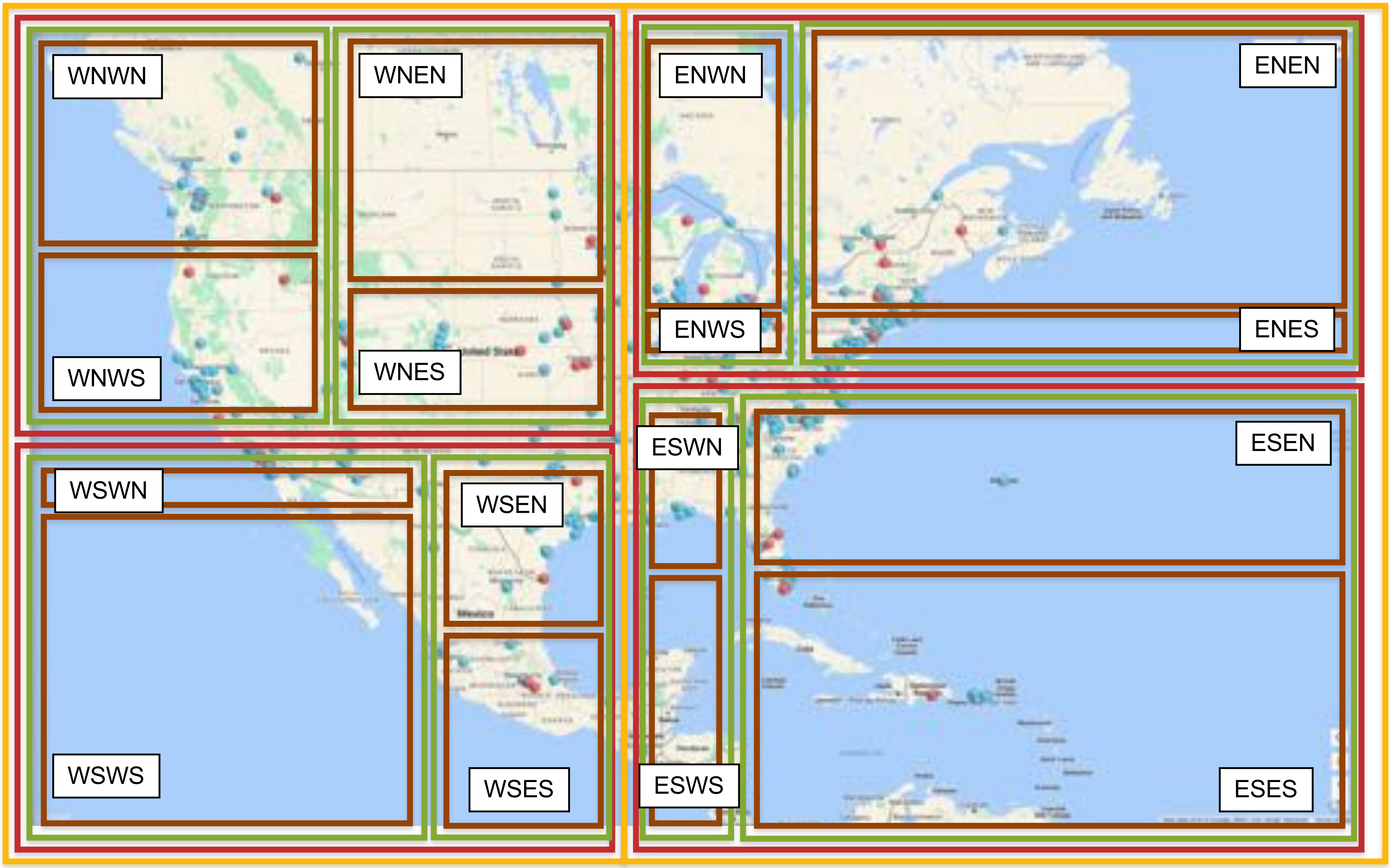
Alternate splits: Each sublist has N/4 element & balanced in both dimensions

Show me the photos!



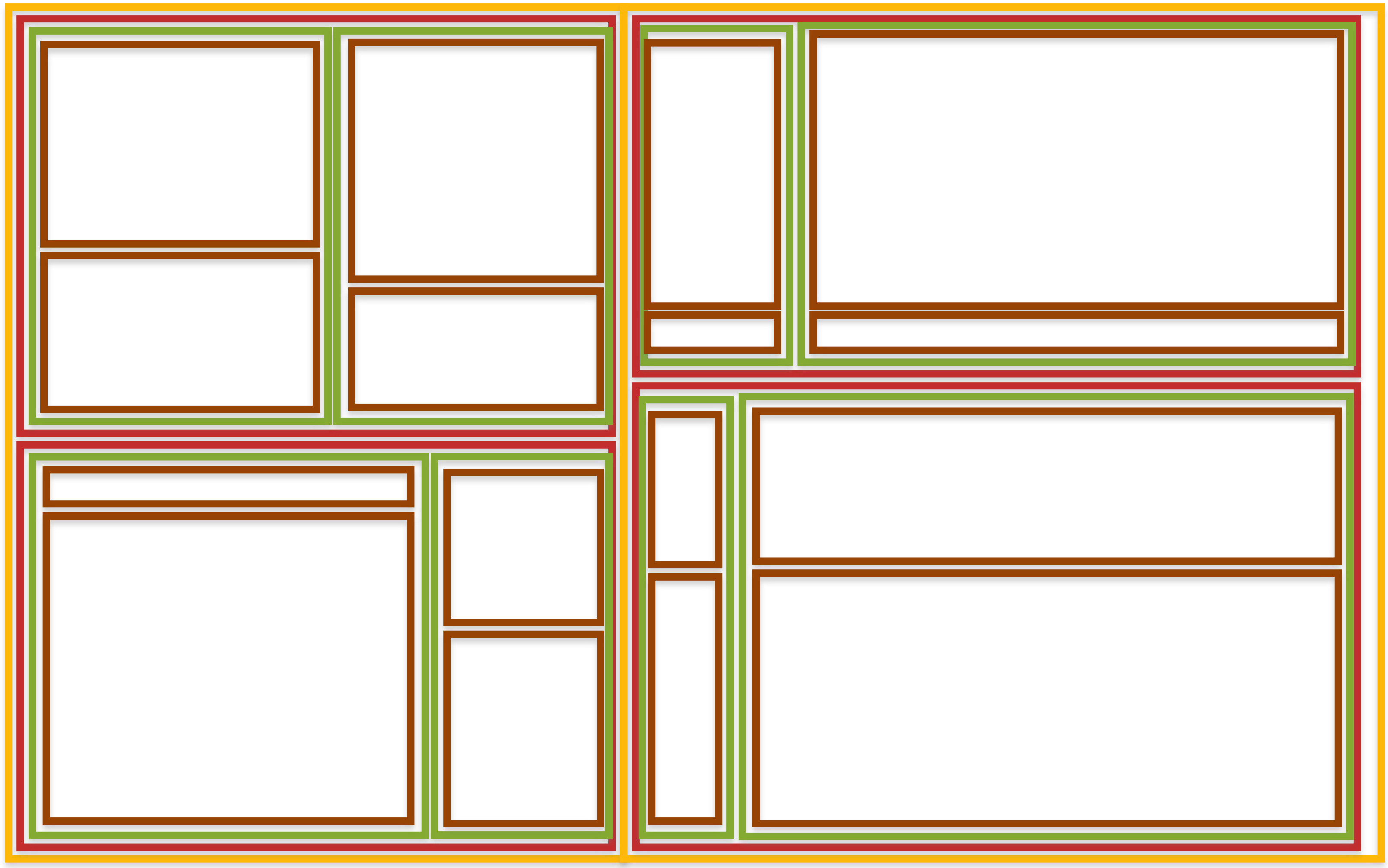
Each sublist has N/8 elements & Balanced in both dimensions

Show me the photos!



Each sublist has N/16 elements & Balanced in both dimensions

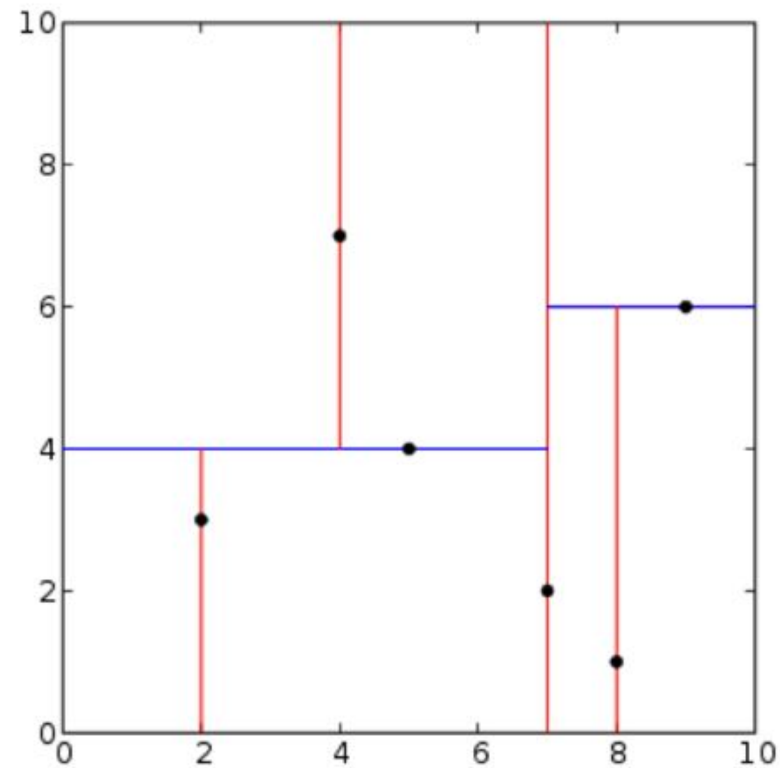
Advanced Data Structure #1: K-d tree



Balanced Binary Search Tree invented by Jon Louis Bentley in 1975
Generalization of the ubiquitous binary search tree
Very fast to build & search almost any type of spatial data

k-d tree pseudocode

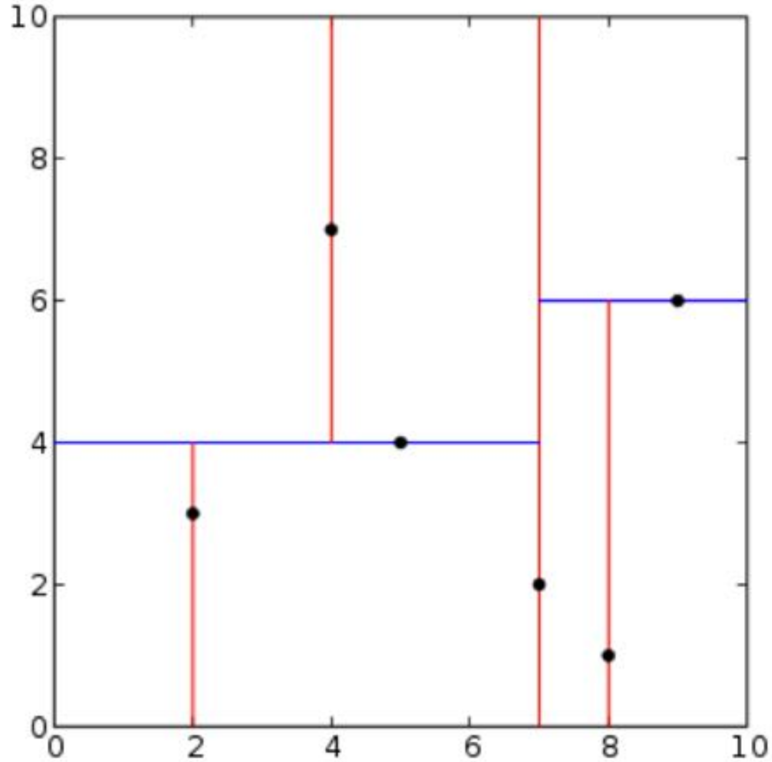
```
Photo getNearest(Point myLoc)
{
  // Region class stores partitions & photos
  Region r = allPhotos

  // While more partitions to go
  while (r.numPhotos() > 1)
  {
    // Partition on Lat/Long
    Dimension d = r.splitDim()

    // Check the relevant coordinate
    if (myLoc.getDim(d) <= r.split)
    {
      // branch to the west/south
      r = r.lo()
    }
    else
    {
      // branch to the east/north
      r = r.hi()
    }
  }

  // just 1 photo, done!
  return r.getPhoto()
}
```



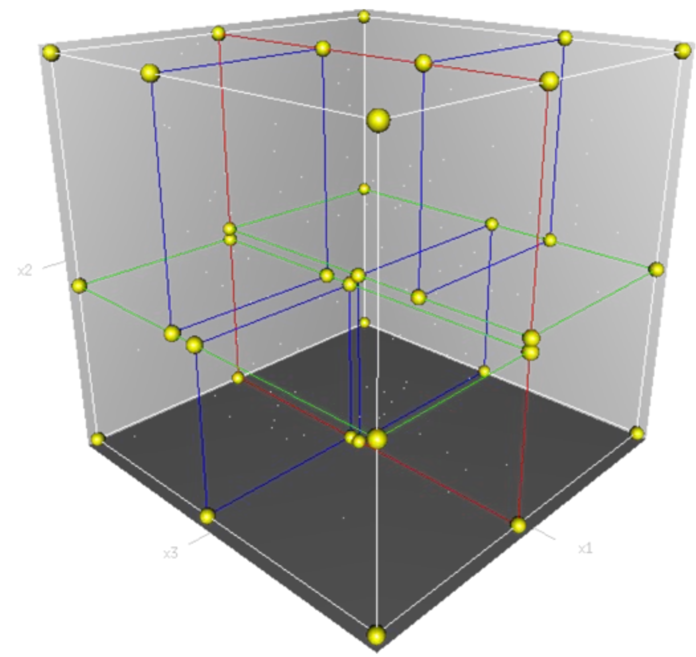
K-d tree data structure to spatially index a large data index the photos

What else might you want to index?

k-d trees in higher dimensions



2d tree:
Alternate left/right, top/bottom



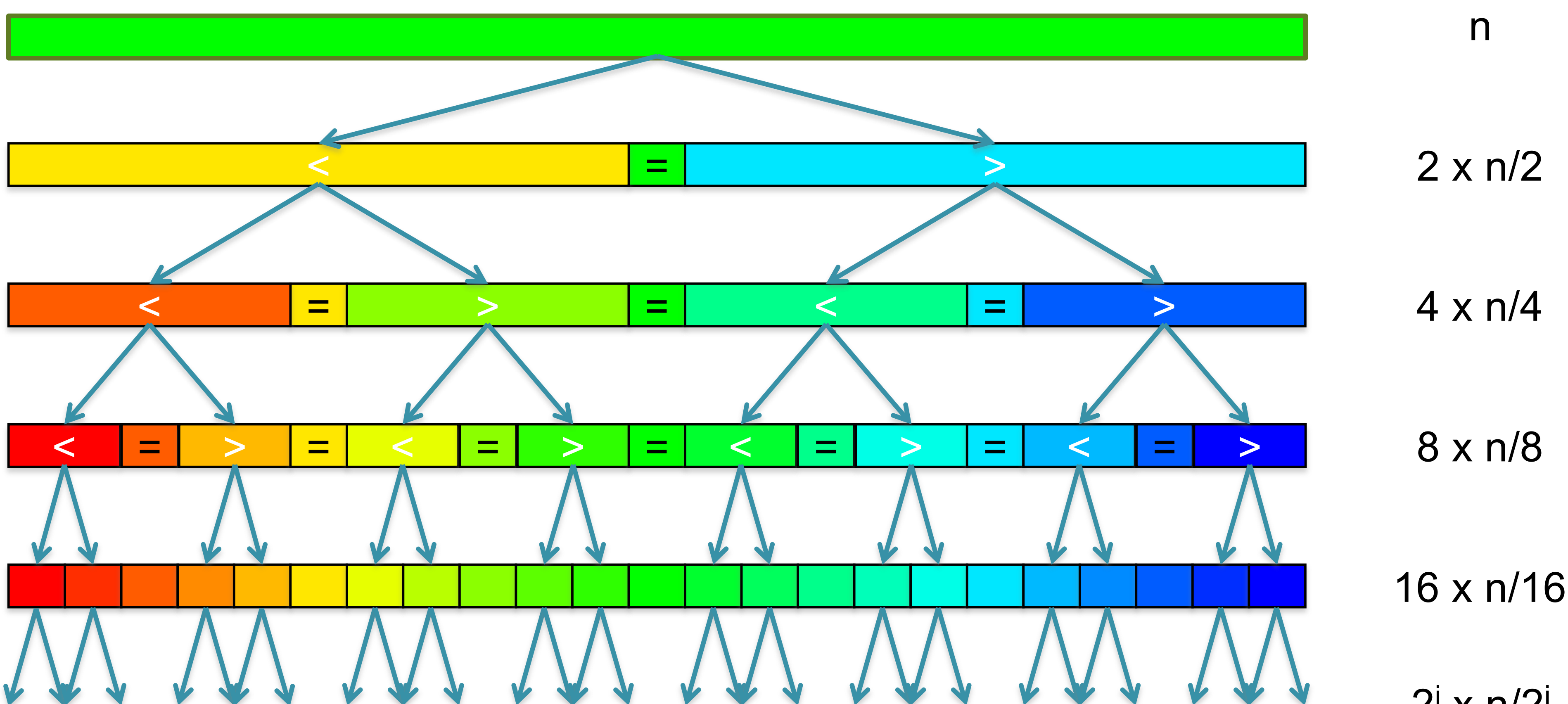
3d tree:
Alternate left/right, top/bottom, up/down

The ‘k’ in k-d tree emphasizes that it works in any number of dimensions
Just gets a little harder to draw for k > 3 ☺

Alternative is to build multiple indices with pointers (URLs) to same set of photos

Divide and Conquer

- Brute force is slow because we have to check every single element
  - How can we split up the unsorted list into independent ranges?
  - Lets recursively split up the elements into greater than/less than range based on the current split line (latitude/longitude)

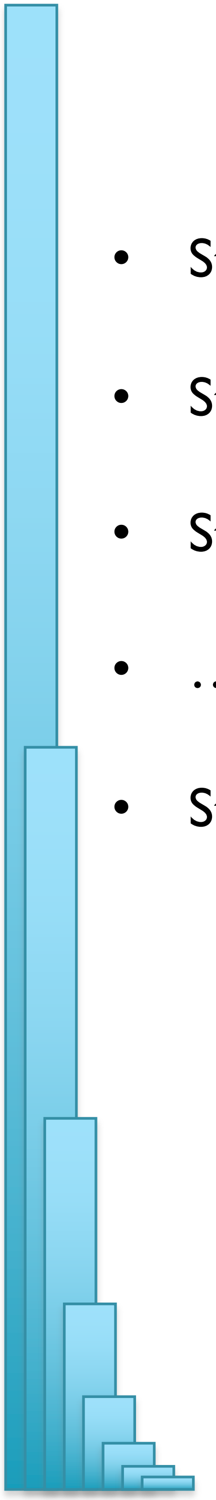


[How many times can we split a list in half?]

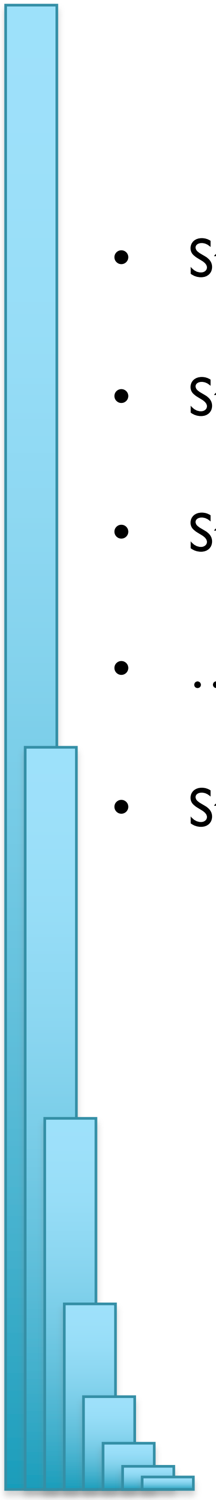
Dividing N in half: 20 Billion

- Step 0: 20,000,000,000 possible elements (N)
- Step 1: 10,000,000,000 possible elements (N/2)
- Step 2: 5,000,000,000 possible elements (N/4)
- …
- Step X: 1 possible element (N/N)

Dividing N in half: 20 Billion



- Step 0: 20,000,000,000 possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000 possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000 possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

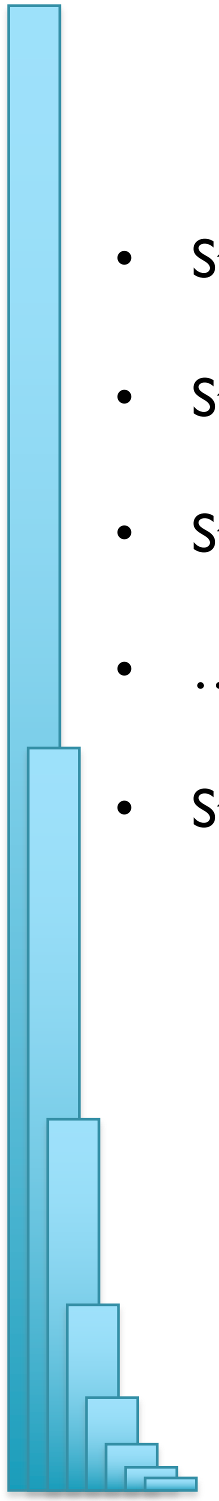


Dividing N in half: 20 Billion

- Step 0: 20,000,000,000 possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000 possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000 possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that: $2^X \geq N$

Dividing N in half: 20 Billion



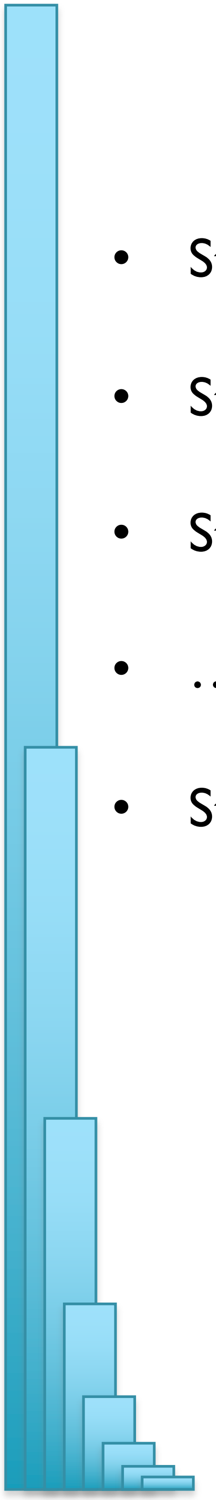
- Step 0: 20,000,000,000 possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000 possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000 possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$$2^X \geq N$$
$$\lg(2^X) \geq \lg(N)$$
$$X \geq \lg(N)$$

$X = ???$

Dividing N in half: 20 Billion



- Step 0: 20,000,000,000 possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000 possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000 possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$$2^X \geq N$$
$$\lg(2^X) \geq \lg(N)$$
$$X \geq \lg(N)$$

X = 35
571.4 million times faster than brute force!

Dividing N in half: 20 TRILLION

- Step 0: 20,000,000,000,**000** possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000,**000** possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000,**000** possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$$2^X \geq N$$
$$\lg(2^X) \geq \lg(N)$$
$$X \geq \lg(N)$$

X = ???

Dividing N in half: 20 TRILLION

- Step 0: 20,000,000,000,**000** possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000,**000** possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000,**000** possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$2^X \geq N$

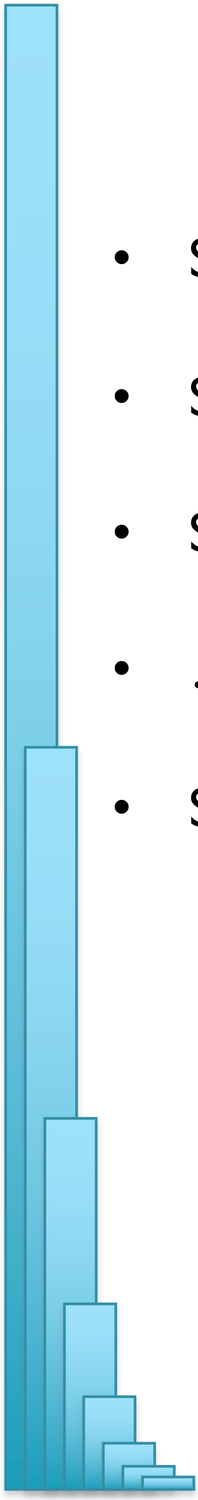
$lg(2^X) \geq lg(N)$

$X \geq lg(N)$

X = 45

571.4 billion times faster than brute force!

Dividing N in half: 20 QUADRILLION



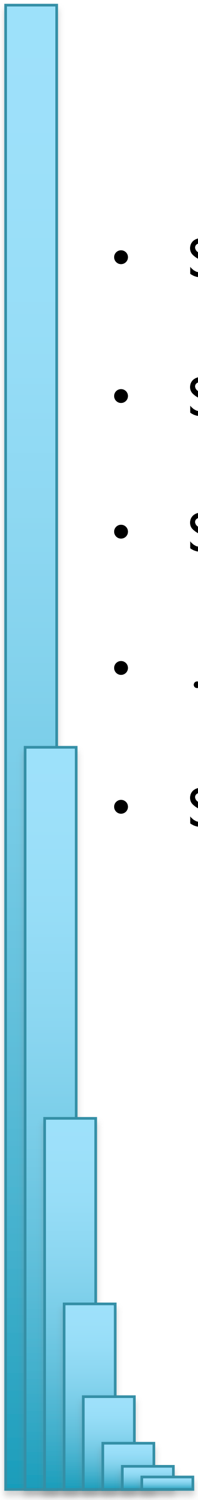
- Step 0: 20,000,000,000,**000,000** possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000,**000,000** possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000,**000,000** possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$$2^X \geq N$$
$$lg(2^X) \geq lg(N)$$
$$X \geq lg(N)$$

X = ???

Dividing N in half: 20 QUADRILLION



- Step 0: 20,000,000,000,**000,000** possible elements ($N/1 = N/2^0$)
- Step 1: 10,000,000,000,**000,000** possible elements ($N/2 = N/2^1$)
- Step 2: 5,000,000,000,**000,000** possible elements ($N/4 = N/2^2$)
- …
- Step X: 1 possible element ($N/N = N/2^X$)

Find X such that:

$$2^X \geq N$$
$$\lg(2^X) \geq \lg(N)$$
$$X \geq \lg(N)$$

X = 55
571.4 trillion times faster than brute force!

How much is a zettabyte?

Unit	Size	~2^x
Byte	1	2^{0}
Kilobyte	1,000	2^{10}
Megabyte	1,000,000	2^{20}
Gigabyte	1,000,000,000	2^{30}
Terabyte	1,000,000,000,000	2^{40}
Petabyte	1,000,000,000,000,000	2^{50}
Exabyte	1,000,000,000,000,000,000	2^{60}
Zettabyte	1,000,000,000,000,000,000,000	2^{70}

How much is a zettabyte?

Unit	Size	~2^x
Byte	1	2^{0}
Kilobyte	1,000	2^{10}
Megabyt		2^{20}
Gigabyte		2^{30}
Terabyte		2^{40}
Petabyte		2^{50}
Exabyte	1,000,000,000,000,000,000	2^{60}
Zettabyte	1,000,000,000,000,000,000,000	2^{70}

For all practical purposes:

$\lg(X) << 70$

Next Steps

1. Reflect on the magic and power of log ☺
2. Register on Piazza
3. Set up Dropbox for yourself!
4. Get comfortable with a editor (VI rules!) and the command line


Welcome to CS 600.226
https://github.com/schatzlab/datastructures2018

Questions?