CS 225

Data Structures

March 4 – AVL Analysis

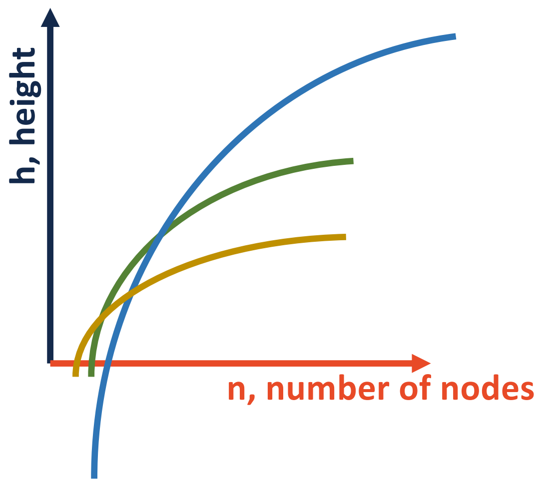
G Carl Evans

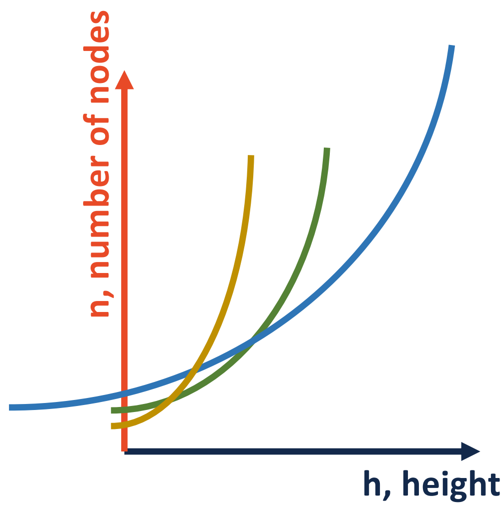
AVL Tree Analysis

We know: insert, remove and find runs in: ___________.

We will argue that: h is __________.

AVL Tree Analysis





- The number of nodes in the tree, $\mathbf{f^{-1}(h)}$, will always be <u>greater than</u> $\mathbf{c \times g^{-1}(h)}$ for all values where $\mathbf{n > k}$.

Plan of Action

Since our goal is to find the lower bound on **n** given **h**, we can begin by defining a function given **h** which describes the smallest number of nodes in an AVL tree of height **h**:

Simplify the Recurrence

N(h) = 1 + N(h - 1) + N(h - 2)

State a Theorem

Theorem: An AVL tree of height h has at least ___________.

Proof:

I. Consider an AVL tree and let **h** denote its height.

II. Case: _______________

An AVL tree of height ____ has at least ____ nodes.

Prove a Theorem

III. Case: _______________

An AVL tree of height ____ has at least ____ nodes.

Prove a Theorem

By an Inductive Hypothesis (IH):

We will show that:

An AVL tree of height ____ has at least ____ nodes.

Prove a Theorem

V. Using a proof by induction, we have shown that:

…and inverting:

Summary of Balanced BST

Red-Black Trees

- Max height: 2 * lg(n)
- Constant number of rotations on insert, remove, and find

AVL Trees

- Max height: 1.44 * lg(n)
- Rotations:

Summary of Balanced BST

Pros:

- Running Time:

  - Improvement Over:

- Great for specific applications:

Summary of Balanced BST

Cons:

- Running Time:

- In-memory Requirement:

Range-based Searches

Q: Consider points in 1D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$.

…what points fall in [11, 42]?

Tree construction:

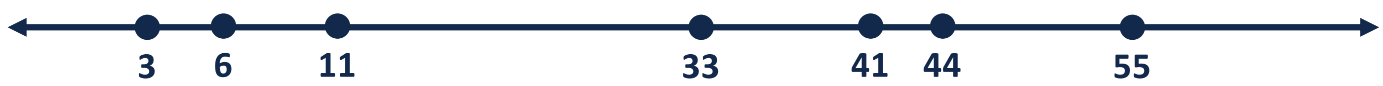
Range-based Searches

Balanced BSTs are useful structures for range-based and nearest-neighbor searches.

Q: Consider points in 1D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$.

…what points fall in [11, 42]?

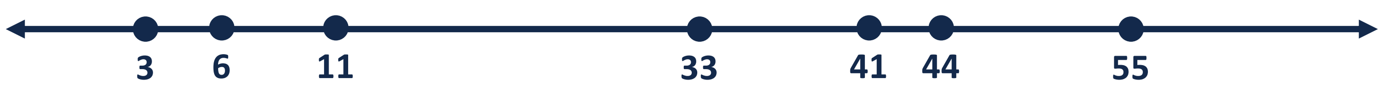
Ex:



Range-based Searches

Q: Consider points in 1D: $\mathbf{p} = \{p_1, p_2, \ldots, p_n\}$.
…what points fall in [11, 42]?

Ex:

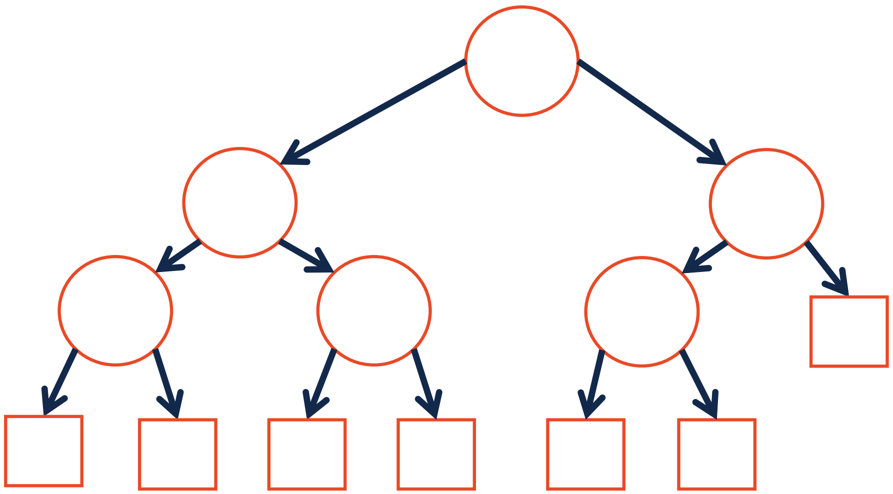


Range-based Searches

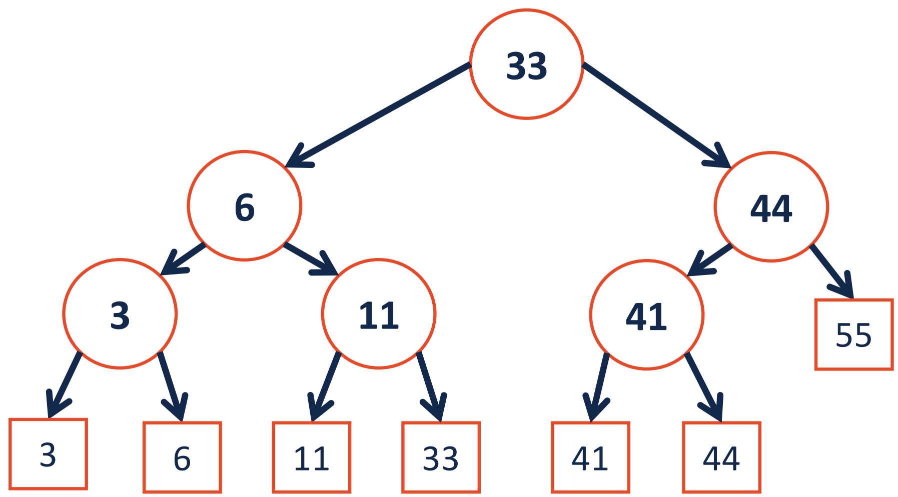
Q: Consider points in 1D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$.
…what points fall in [11, 42]?

Tree construction:

Range-based Searches



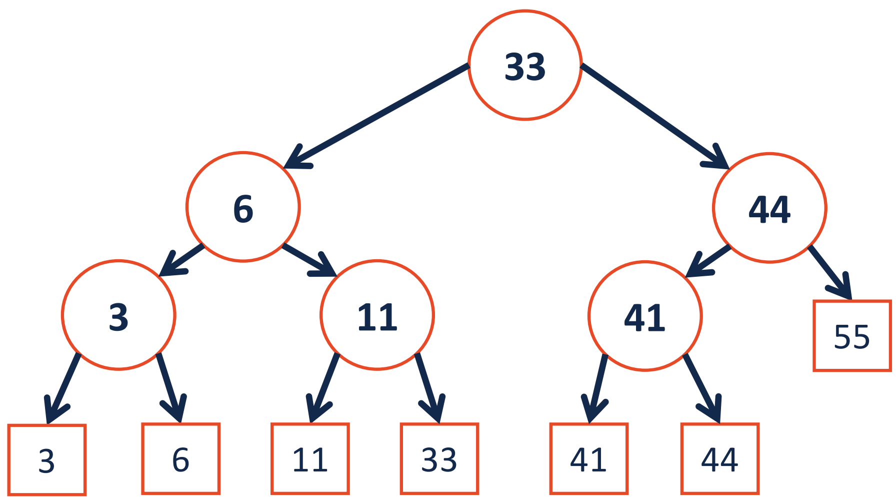
Range-based Searches



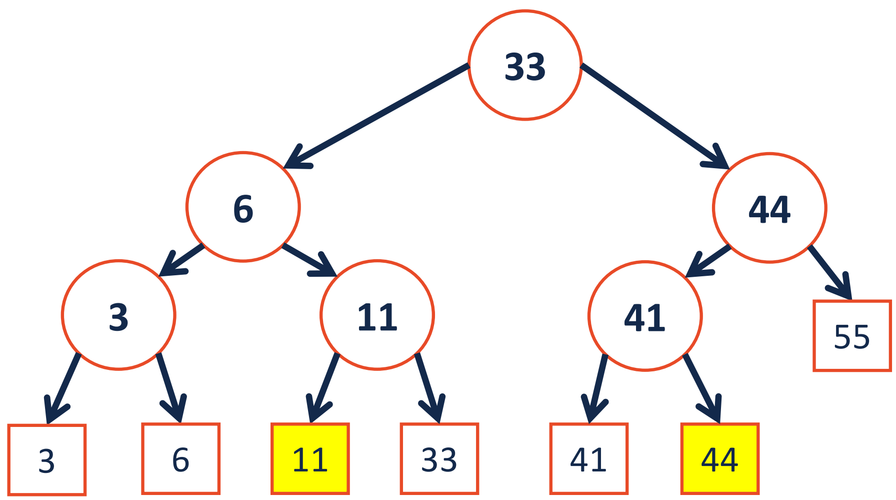
Range-based Searches

Q: Consider points in 1D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$.

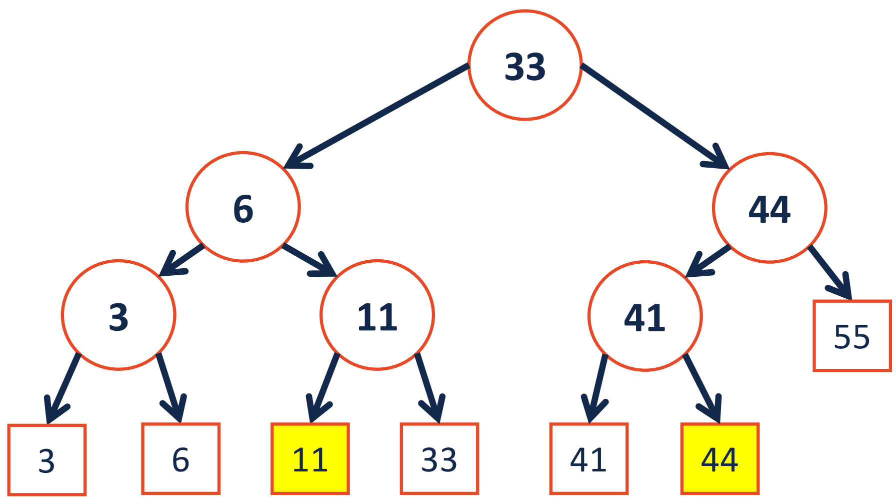
…what points fall in [11, 42]?



Range-based Searches



Running Time

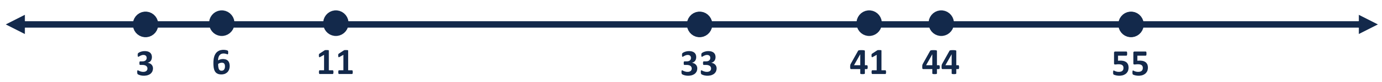


Range-based Searches

Q: Consider points in 1D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$.

…what points fall in [11, 42]?

Ex:



Red-Black Trees in C++

C++ provides us a balanced BST as part of the standard library:

```
std::map<K, V>
```

```
V & std::map<K, V>::operator[](const K & )
```

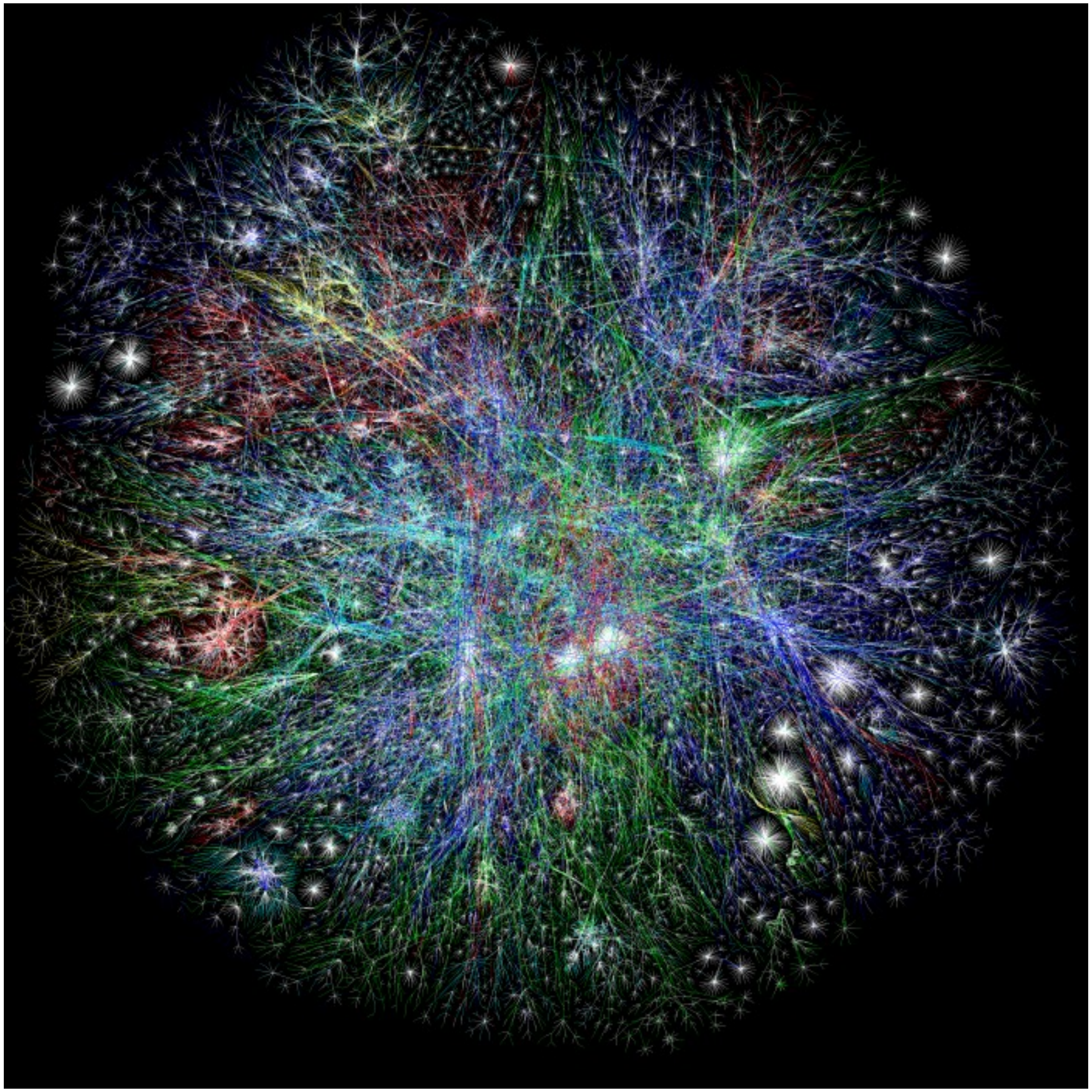
```
iterator std::map<K, V>::lower_bound( const K & )
iterator std::map<K, V>::upper_bound( const K & )
```

Every Data Structure So Far

	Unsorted Array	Sorted Array	Unsorted List	Sorted List	Binary Tree	BST	AVL
Find							
Insert							
Remove							
Traverse							

CS 225 Final Project

Working with data and using graphs

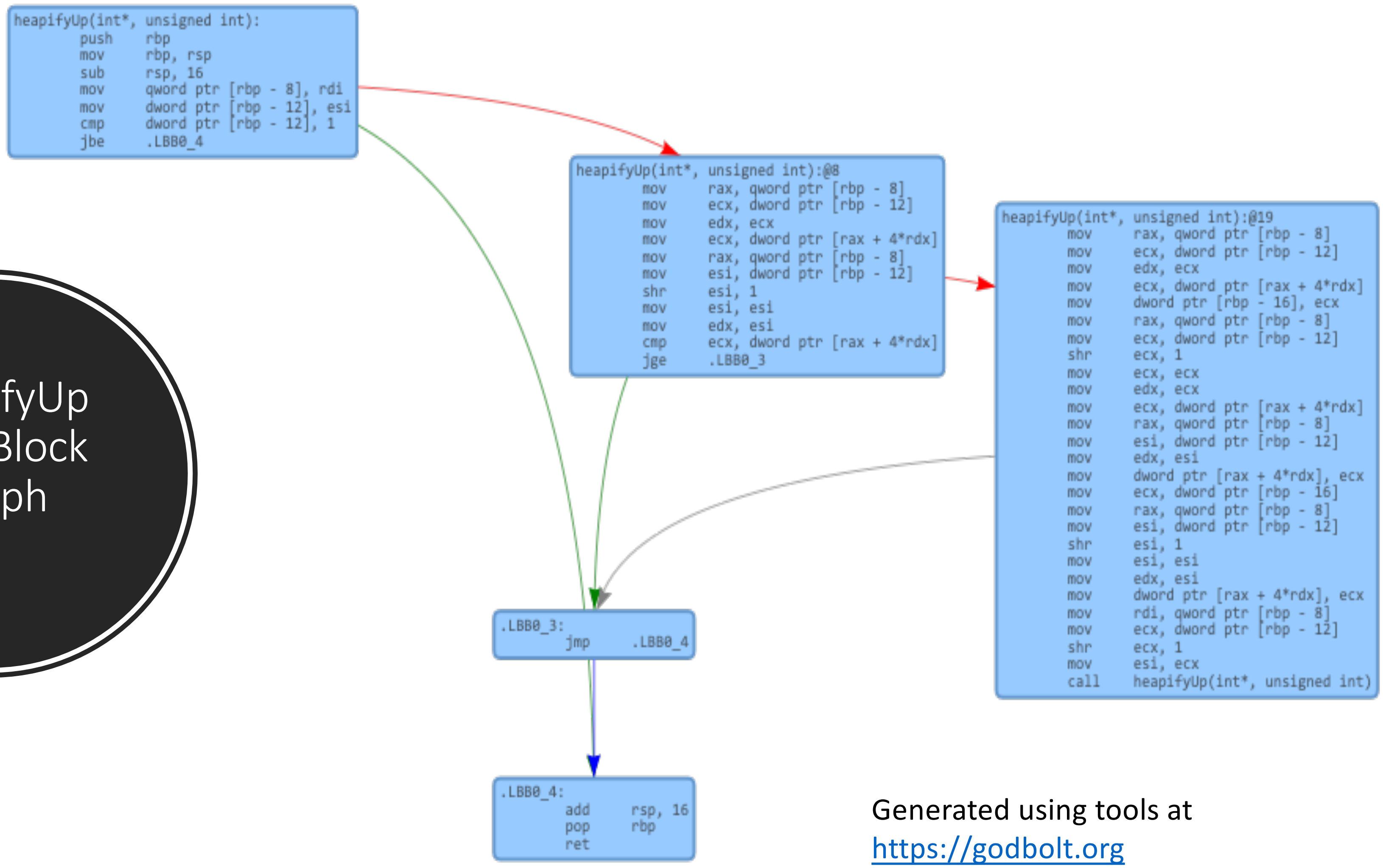


The Internet 2003

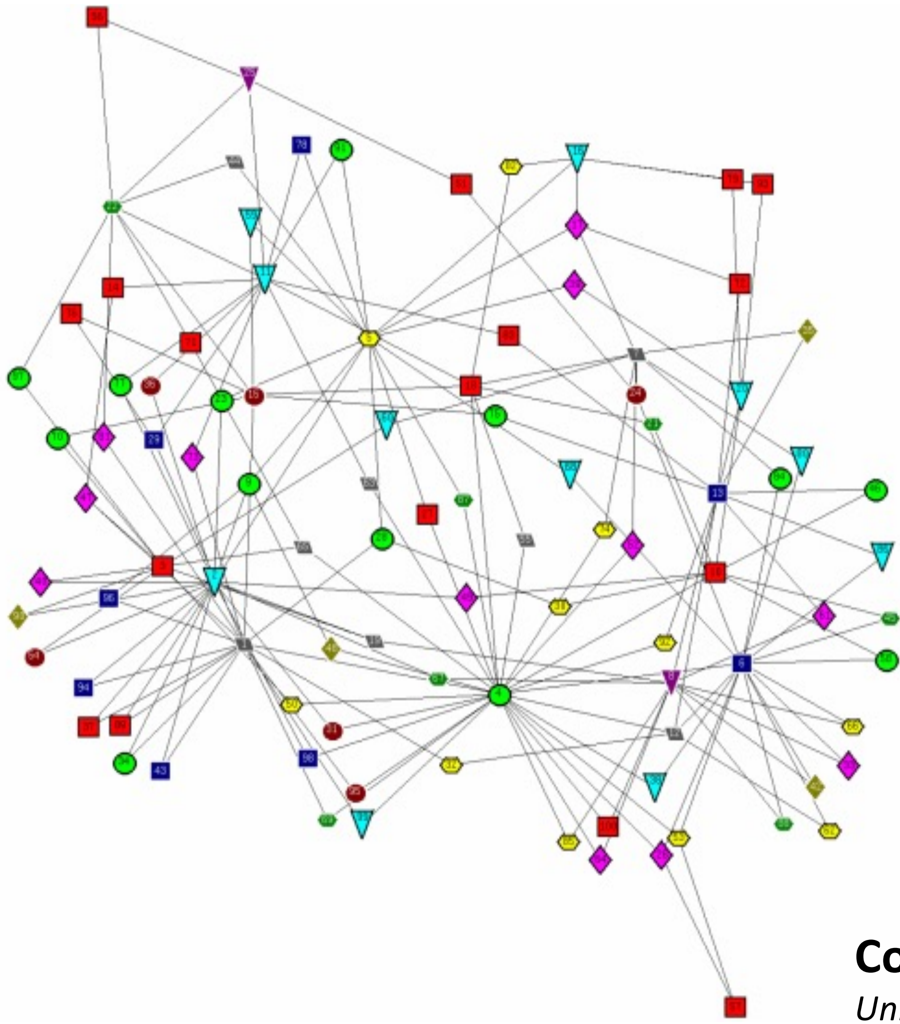
The OPTE Project (2003)

Map of the entire internet; nodes are routers; edges are connections.

HeapifyUp BasicBlock Graph



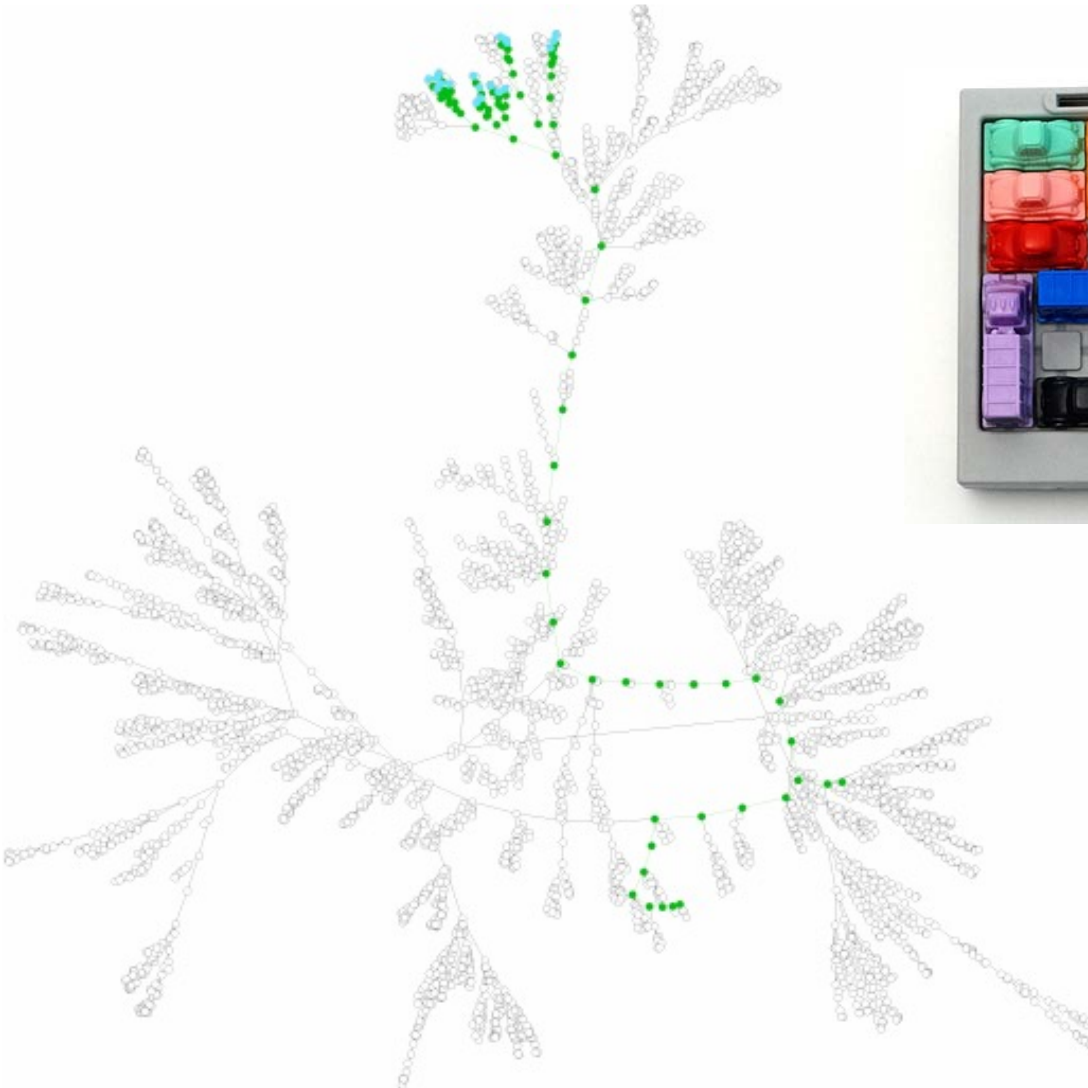
Generated using tools at
https://godbolt.org



Conflict-Free Final Exam Scheduling Graph

Unknown Source

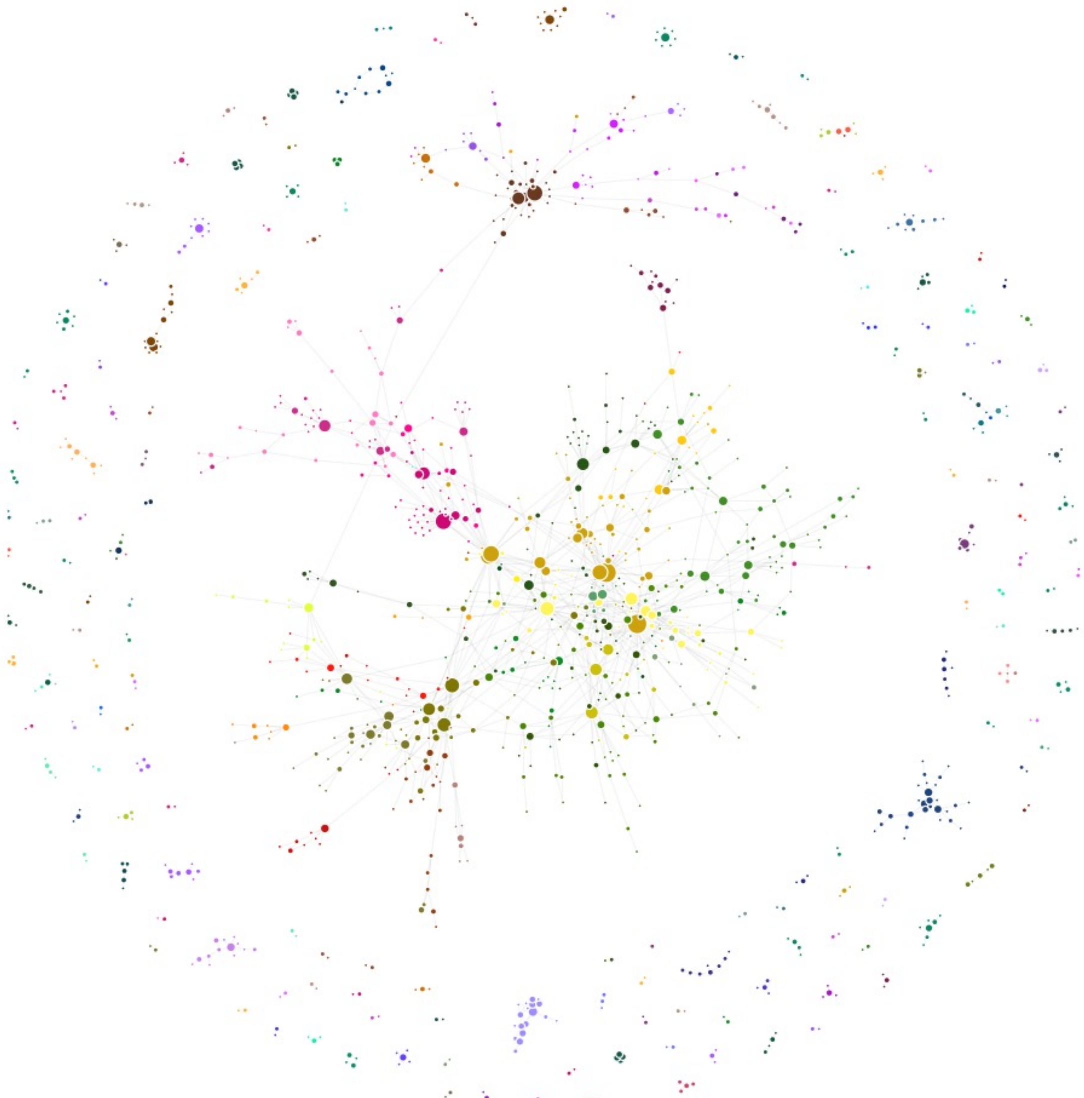
Presented by Cinda Heeren, 2016



“Rush Hour” Solution

Unknown Source

Presented by Cinda Heeren, 2016

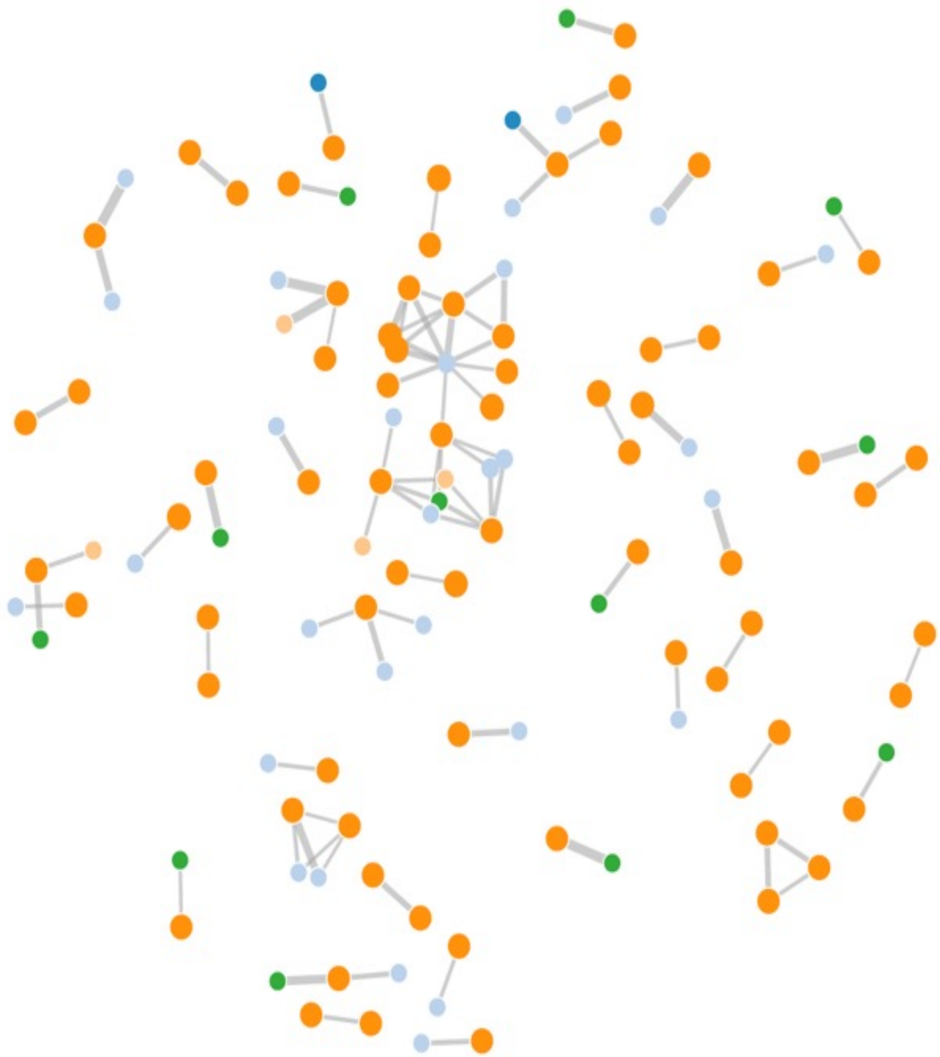


Class Hierarchy At University of Illinois Urbana-Champaign

A. Mori, W. Fagen-Ulmschneider, C. Heeren

Graph of every course at UIUC; nodes are courses, edges are prerequisites

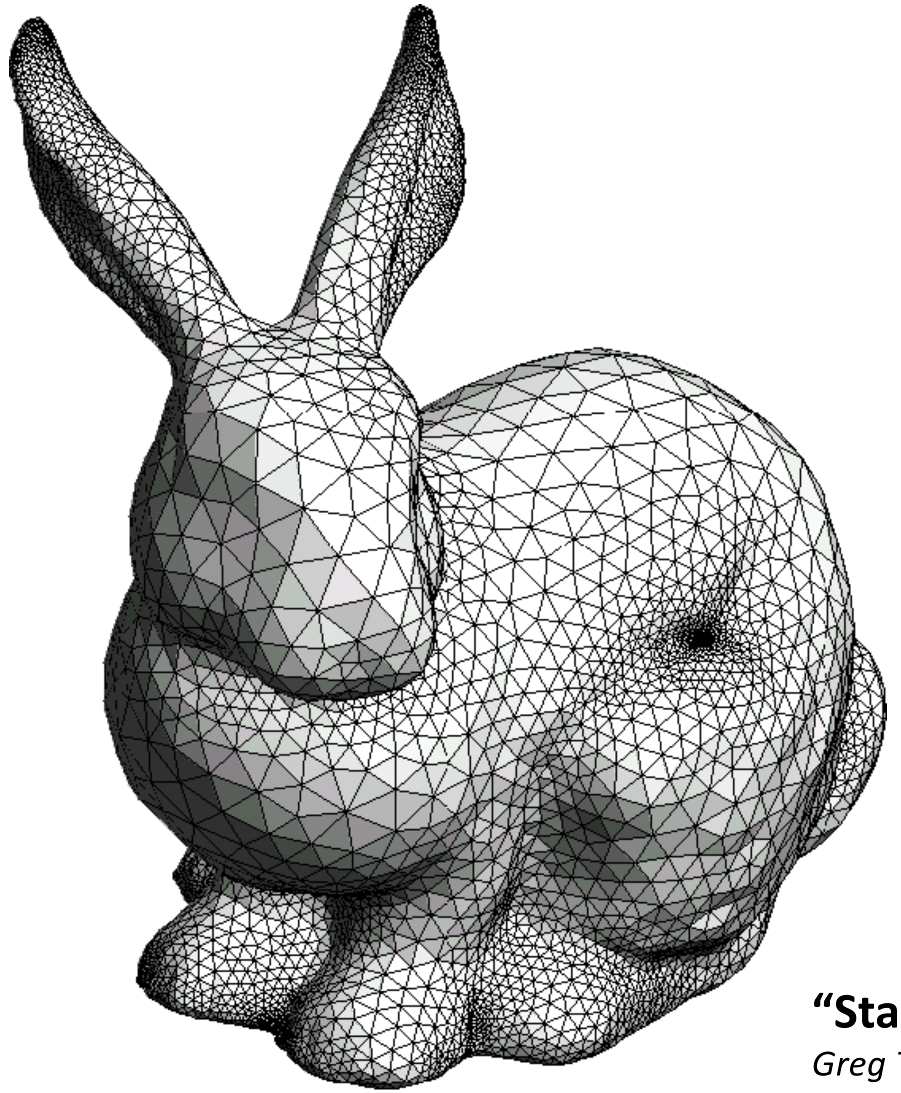
http://waf.cs.illinois.edu/discovery/class_hierarchy_at_illinois/



MP Collaborations in CS 225

Unknown Source

Presented by Cinda Heeren, 2016



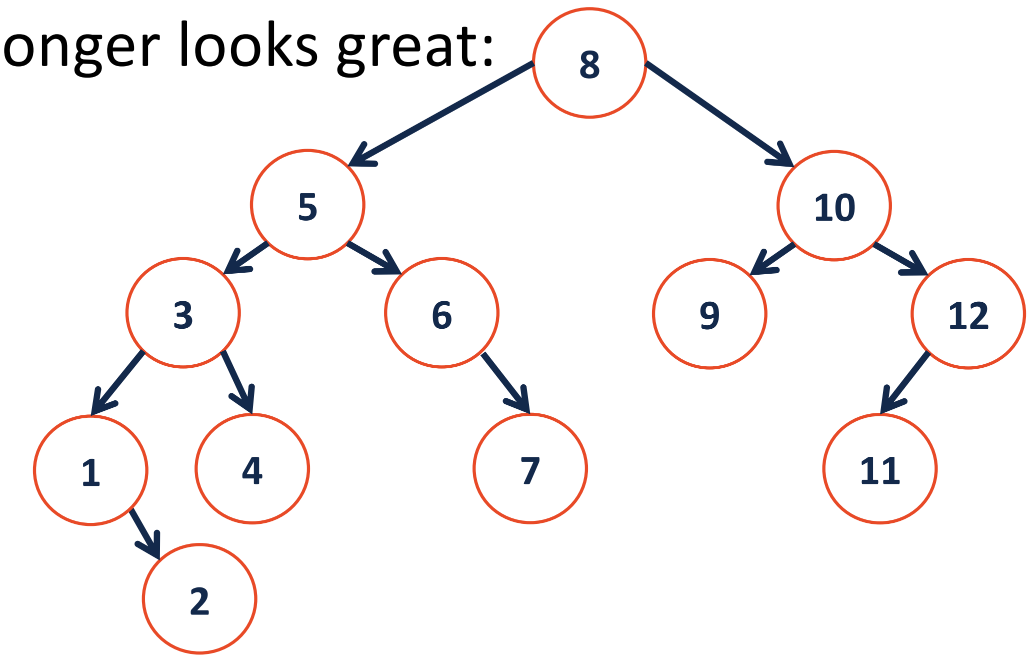
“Stanford Bunny”
Greg Turk and Mark Levoy (1994)

B-Tree Motivation

In Big-O we have assumed uniform time for all operations, but this isn't always true.

However, seeking data from the cloud may take 40ms+.

...an O(lg(n)) AVL tree no longer looks great:



BTree Motivations

Knowing that we have large seek times for data, we want to: