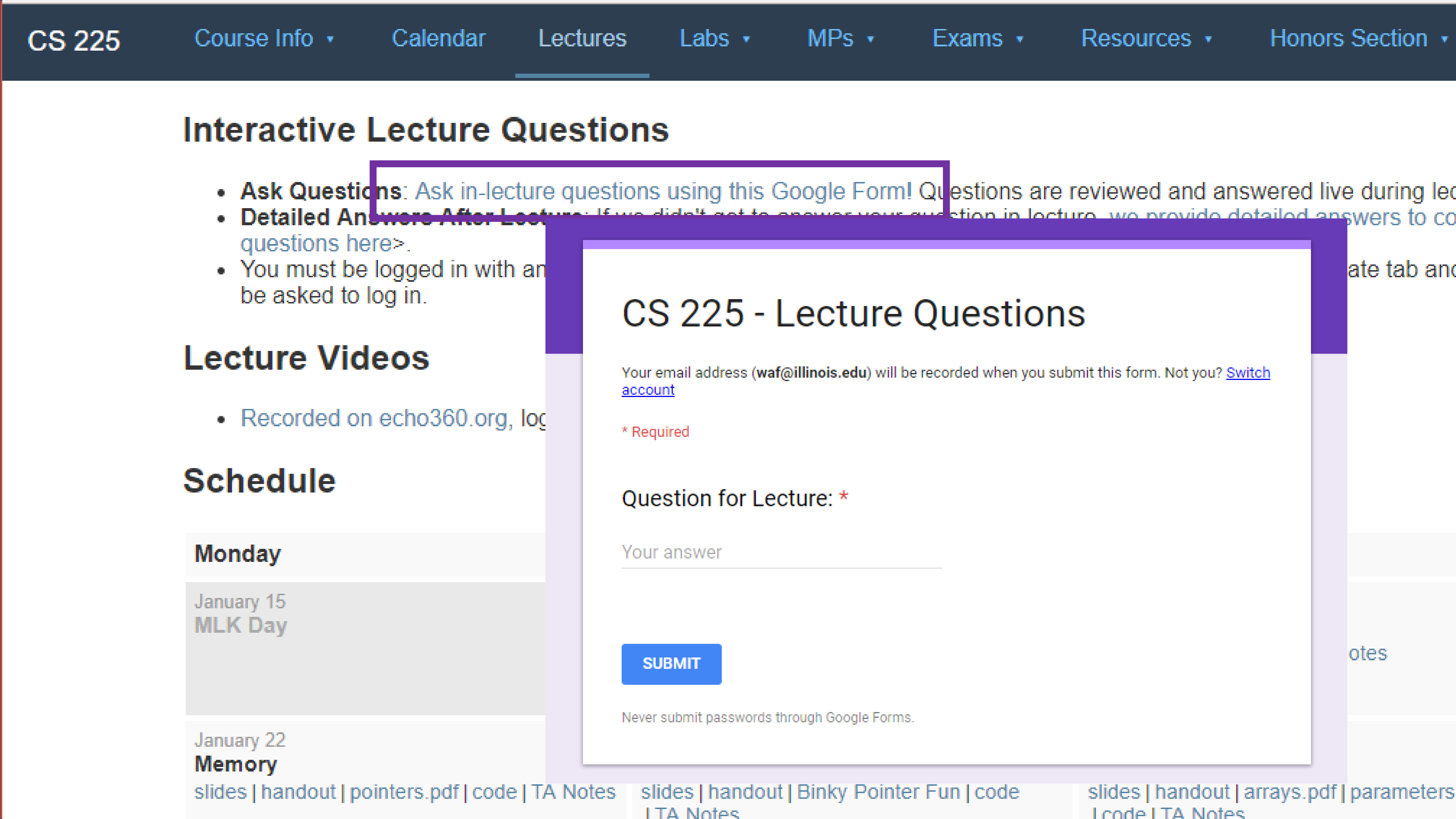
CS 225

Data Structures

Feb. 23 – BST Remove

Wade Fagen-Ulmschneider



Interactive Lecture Questions

- **Ask Questions**: Ask in-lecture questions using this Google Form! Questions are reviewed and answered live during lec
- **Detailed Answers After Lecture**: If we didn't get to answer your question in lecture, we provide detailed answers to co questions here>.
- You must be logged in with an ate tab and be asked to log in.

Lecture Videos

- Recorded on echo360.org, log

Schedule

Monday		
January 15 **MLK Day**		otes
January 22 **Memory** slides | handout | pointers.pdf | code | TA Notes	slides | handout | Binky Pointer Fun | code | TA Notes	slides | handout | arrays.pdf | parameters | code | TA Notes

CS 225 - Lecture Questions

Your email address (**waf@illinois.edu**) will be recorded when you submit this form. Not you? Switch account

* Required

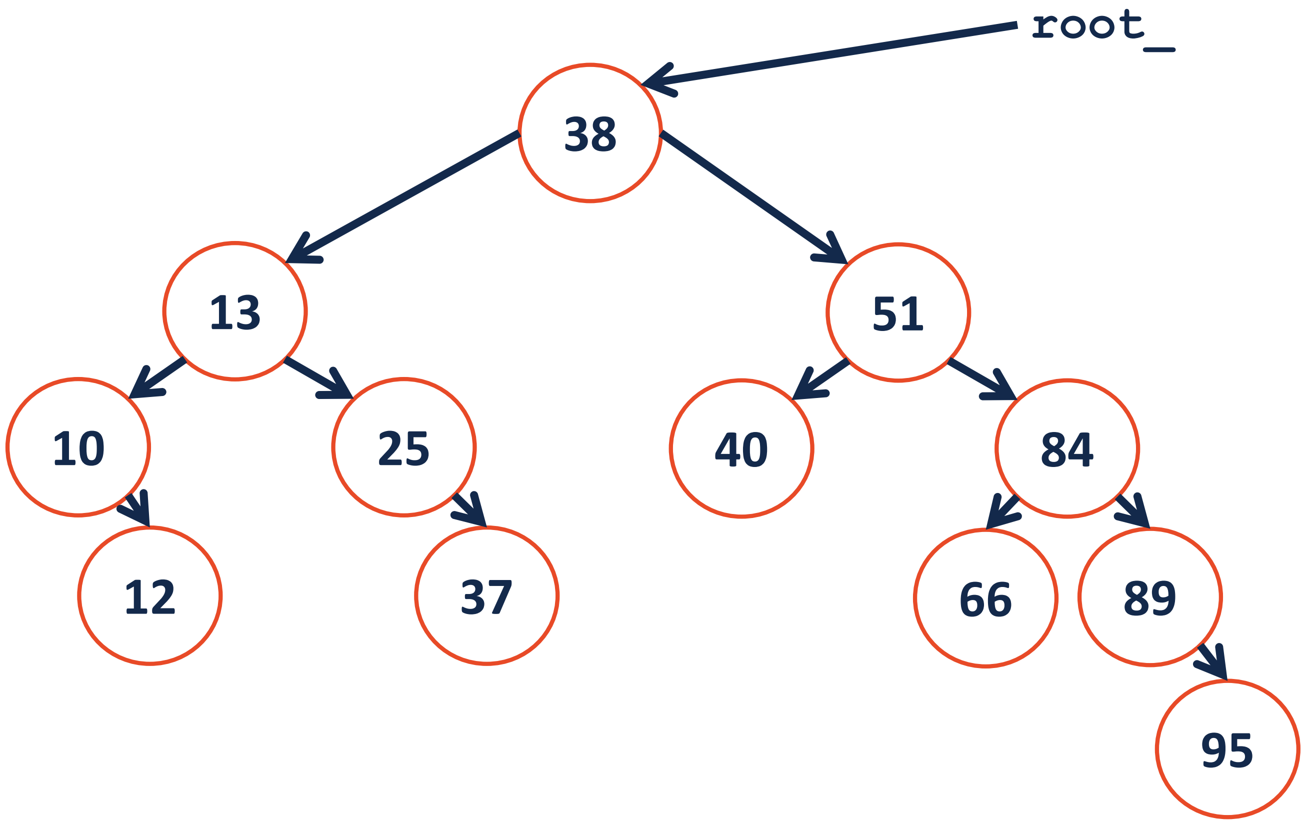
Question for Lecture: *

Your answer

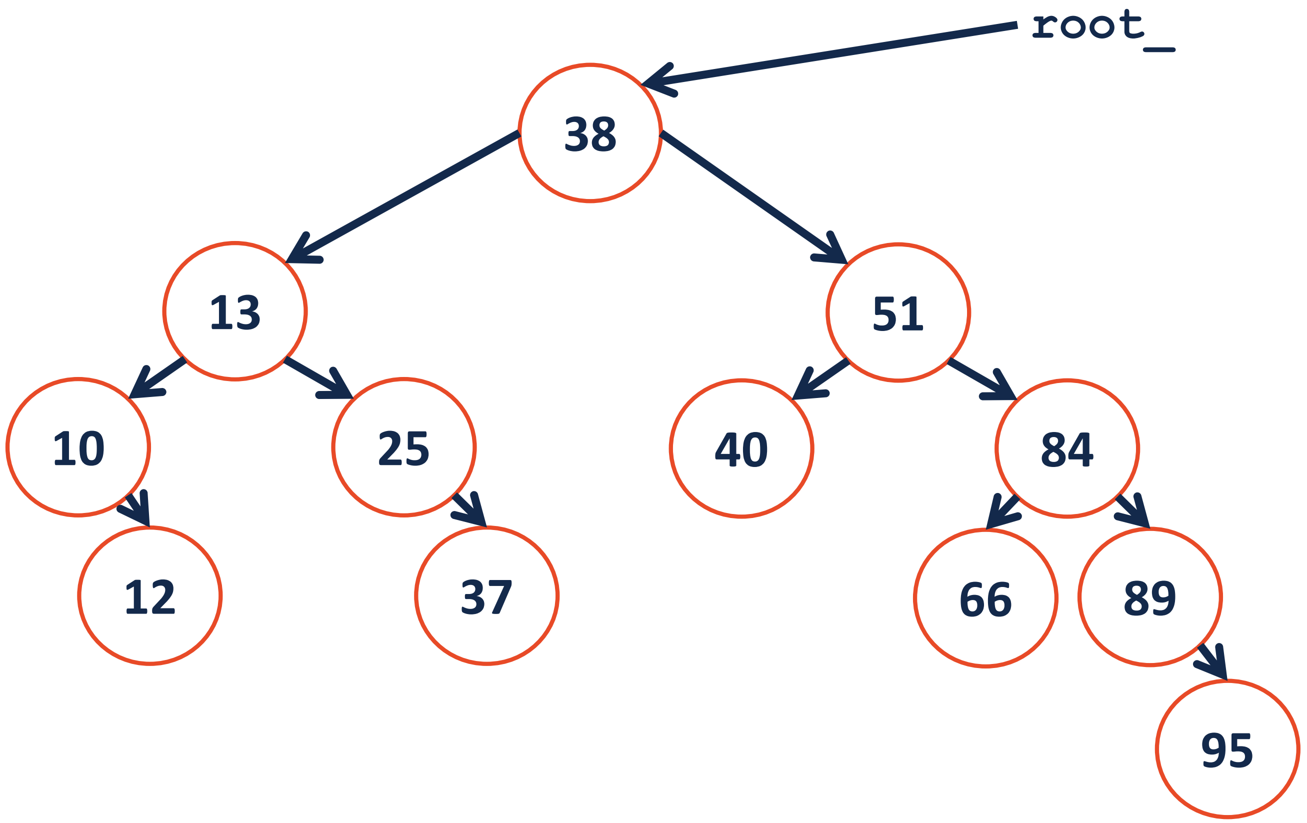
SUBMIT

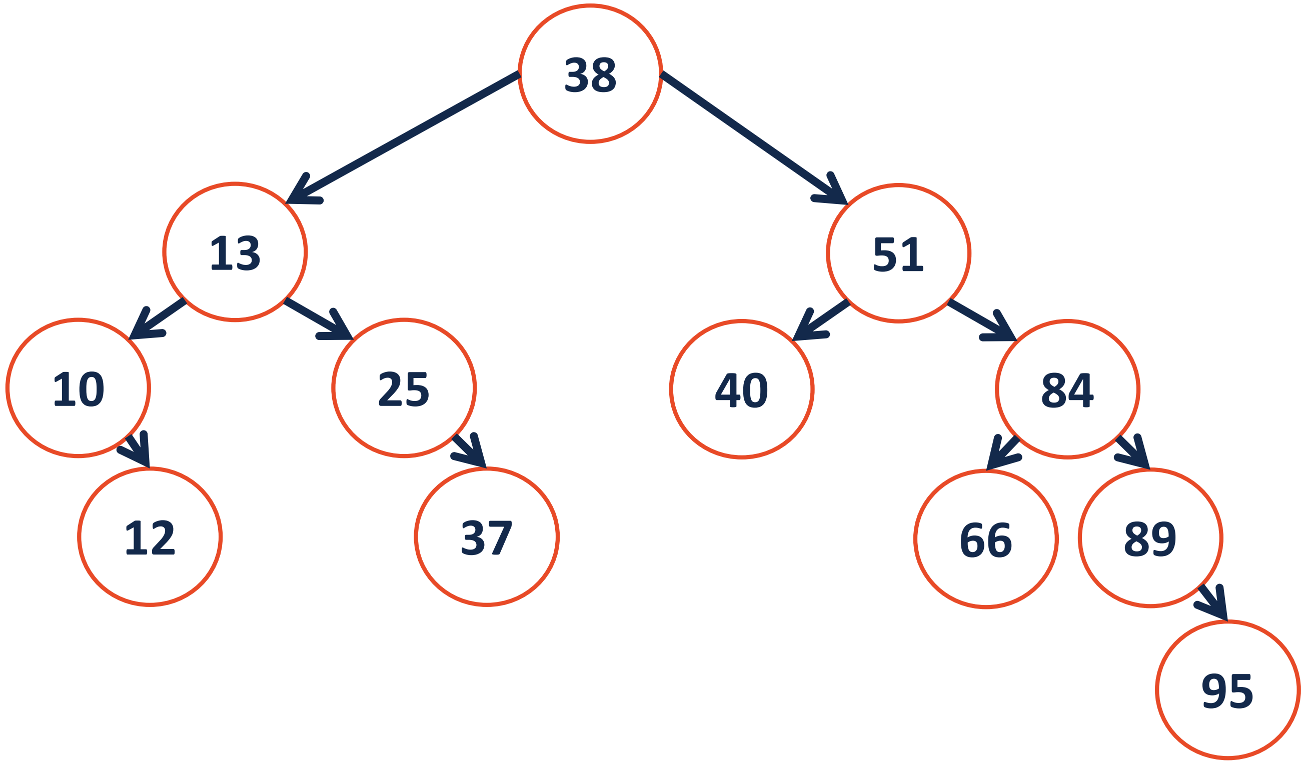
Never submit passwords through Google Forms.

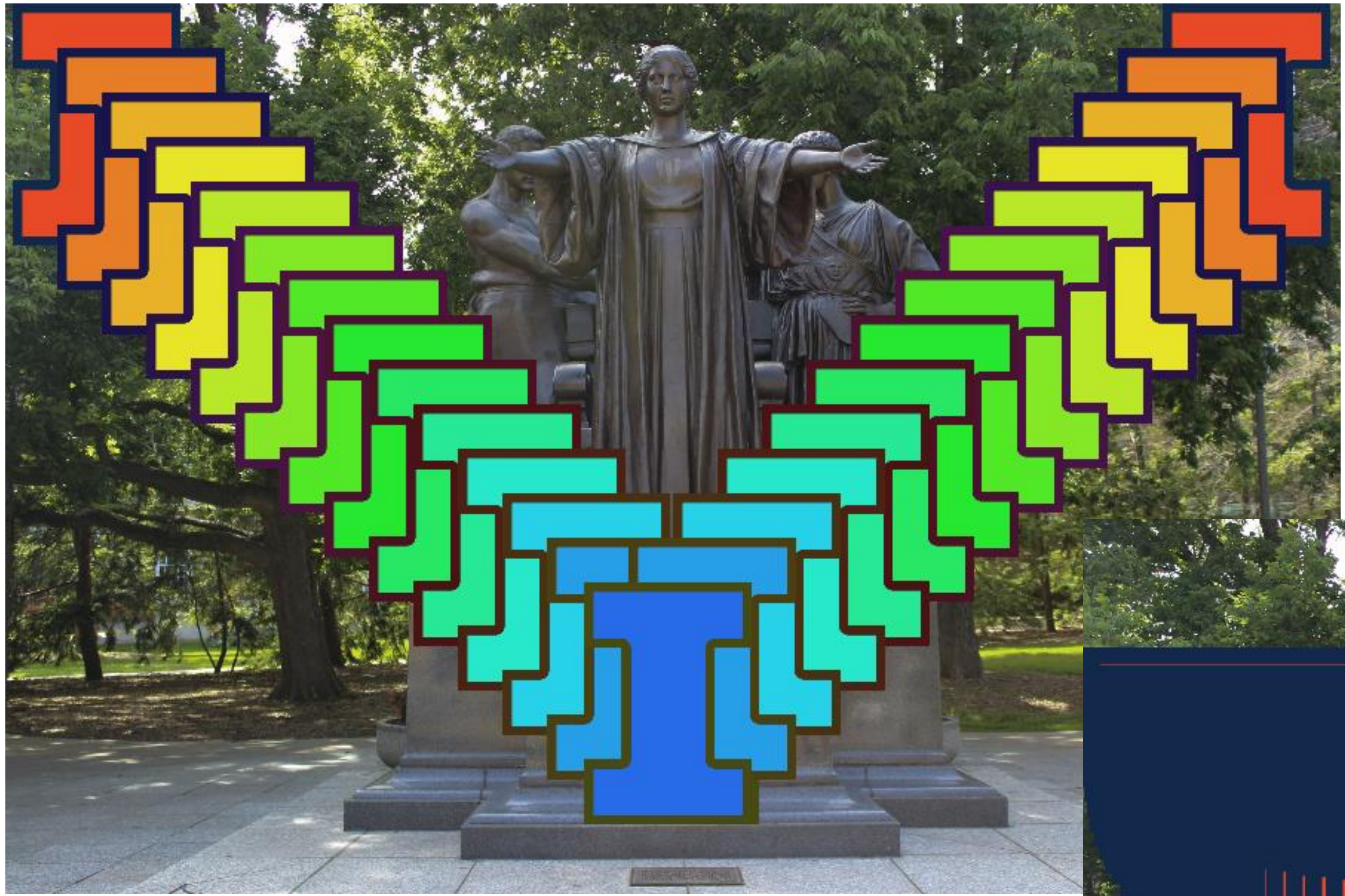
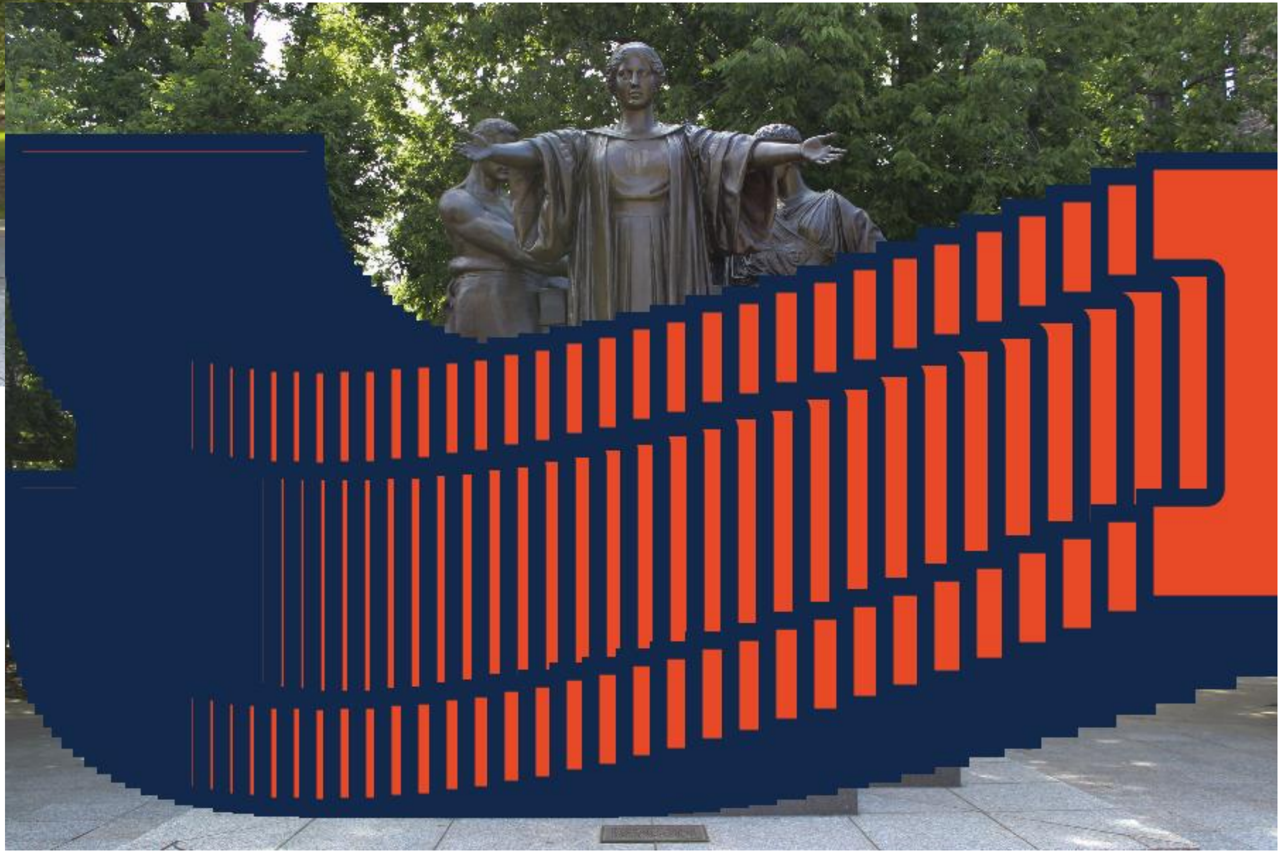
```
template<typename K, typename V>

void BST::_insert(TreeNode *& root, K & key, V & value) {
  TreeNode *t = _find(root, key);
  t = new TreeNode(key, value);
}
```



```
template<typename K, typename V>

void BST::_insert(TreeNode *& root, K & key, V & value) {
  TreeNode *t = _find(root, key);
  t = new TreeNode(key, value);
}
```

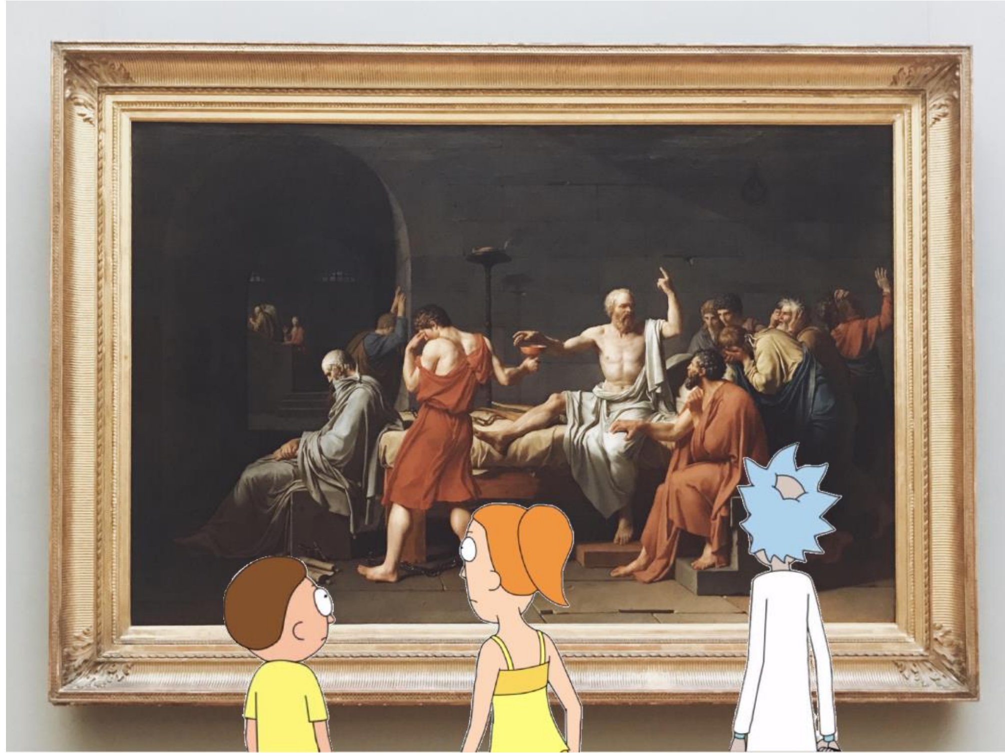


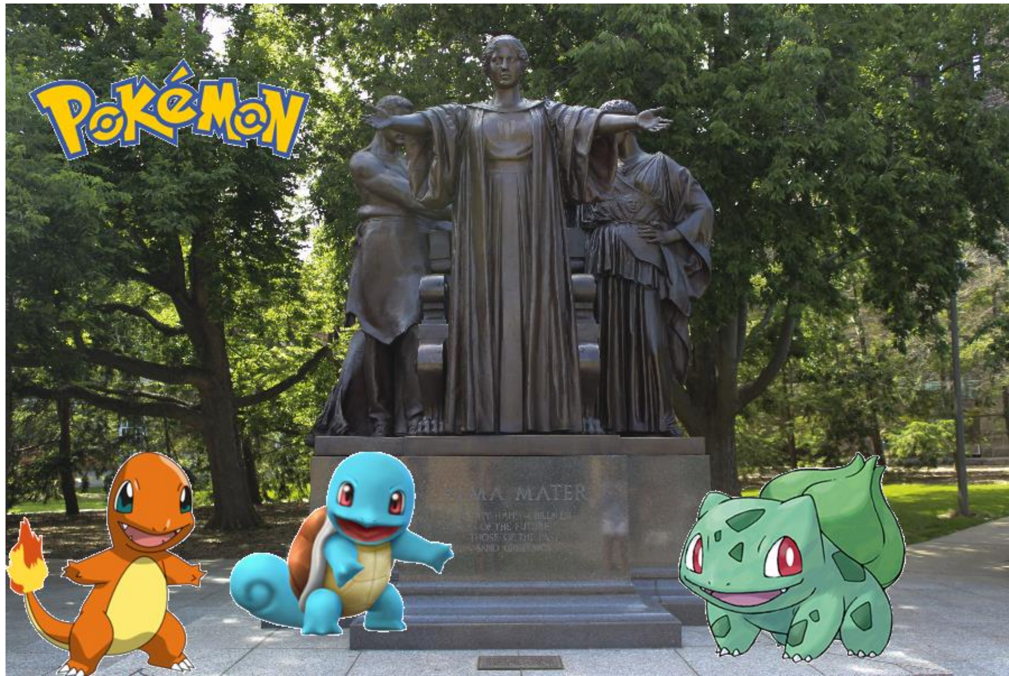

38
13
51
10
25
40
84
12
37
66
89
95

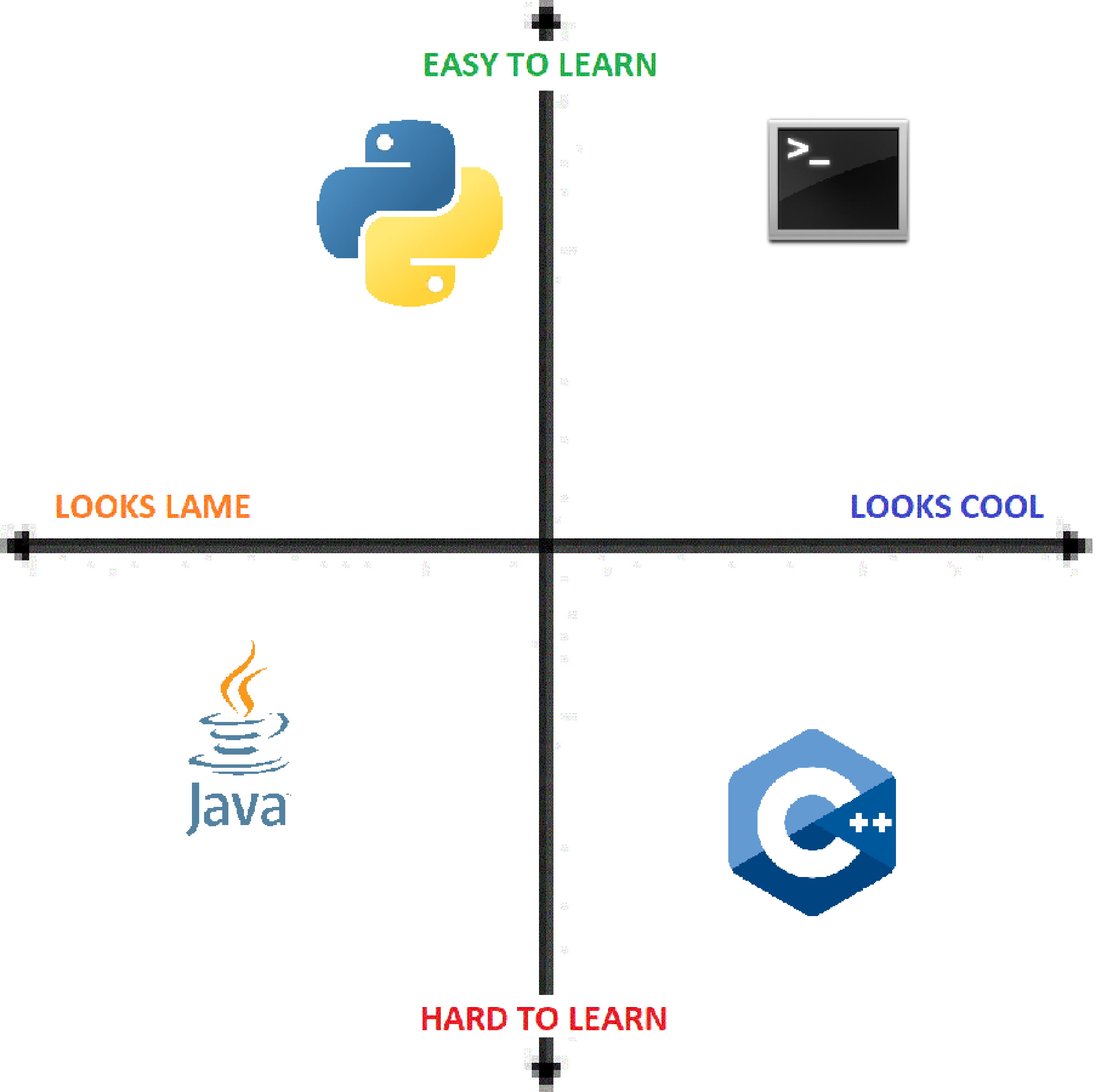



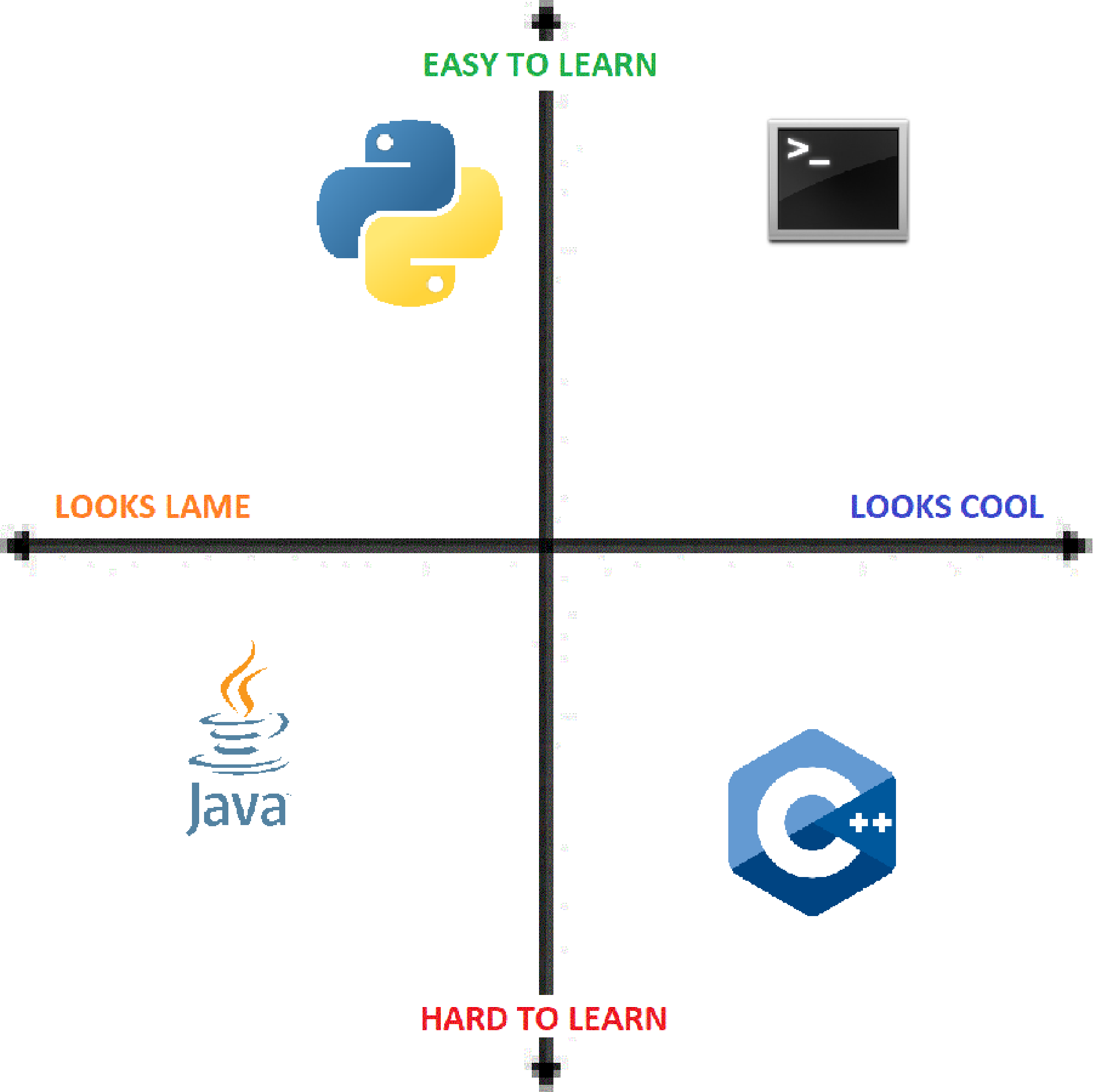
HAPPY
Valentine's
DAY

LED ZEPPELIN
PINK FLOYD
LONG LIVE ROCK 'N' ROLL
RADIOHEAD
AC/DC
EAGLES
hell freezes over

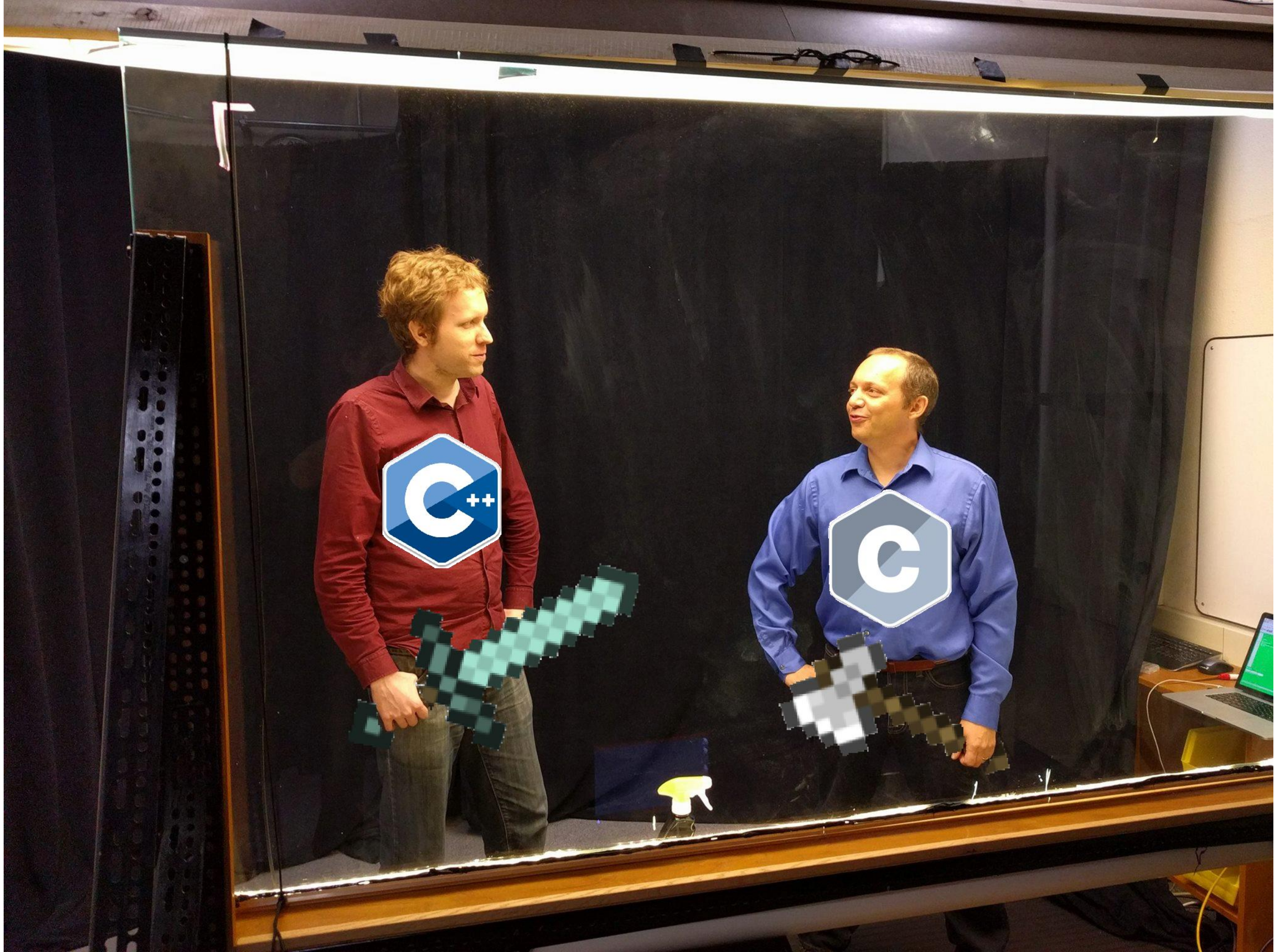



Pokémon
MA MATER


EASY TO LEARN
LOOKS LAME
LOOKS COOL
HARD TO LEARN
>_
Java


EASY TO LEARN
LOOKS LAME
LOOKS COOL
Java
HARD TO LEARN

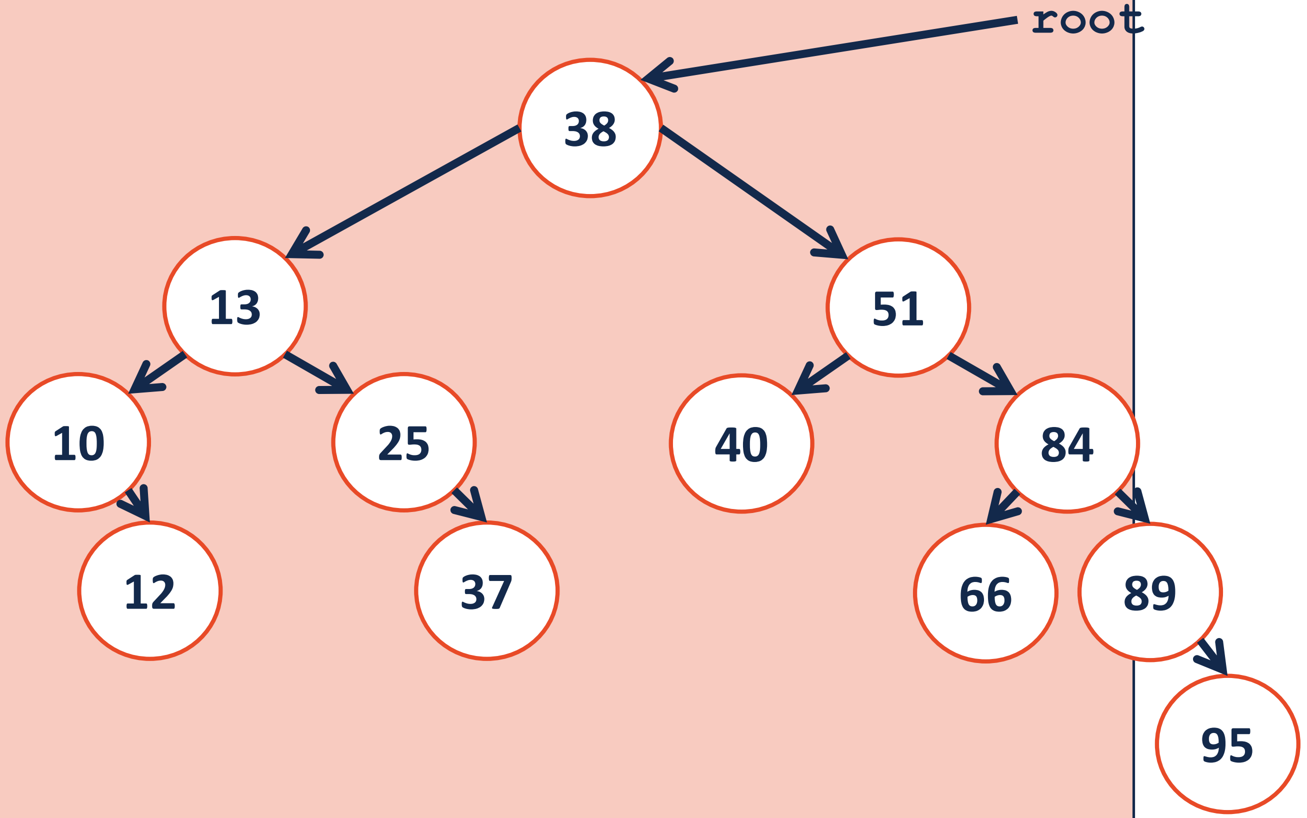
Creating memes with photoshop
Creating memes with microsoft word
Creating memes with MS paint
Creating memes with LibreOffice
Creating memes with
invest in clout-coin

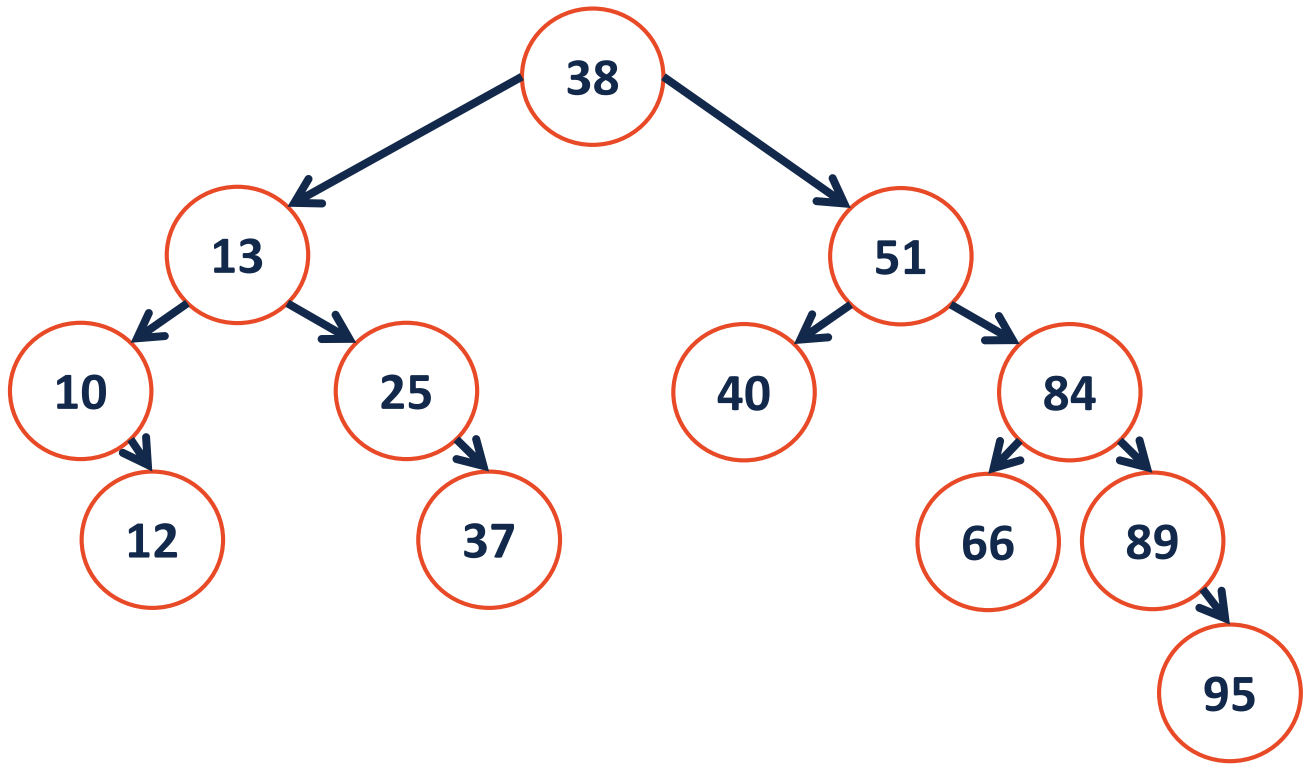

C++
C

Tonight's the night when we forget about
the deadlines, it's time, uh oh
I don't know about you, but
I'm FAILING 225

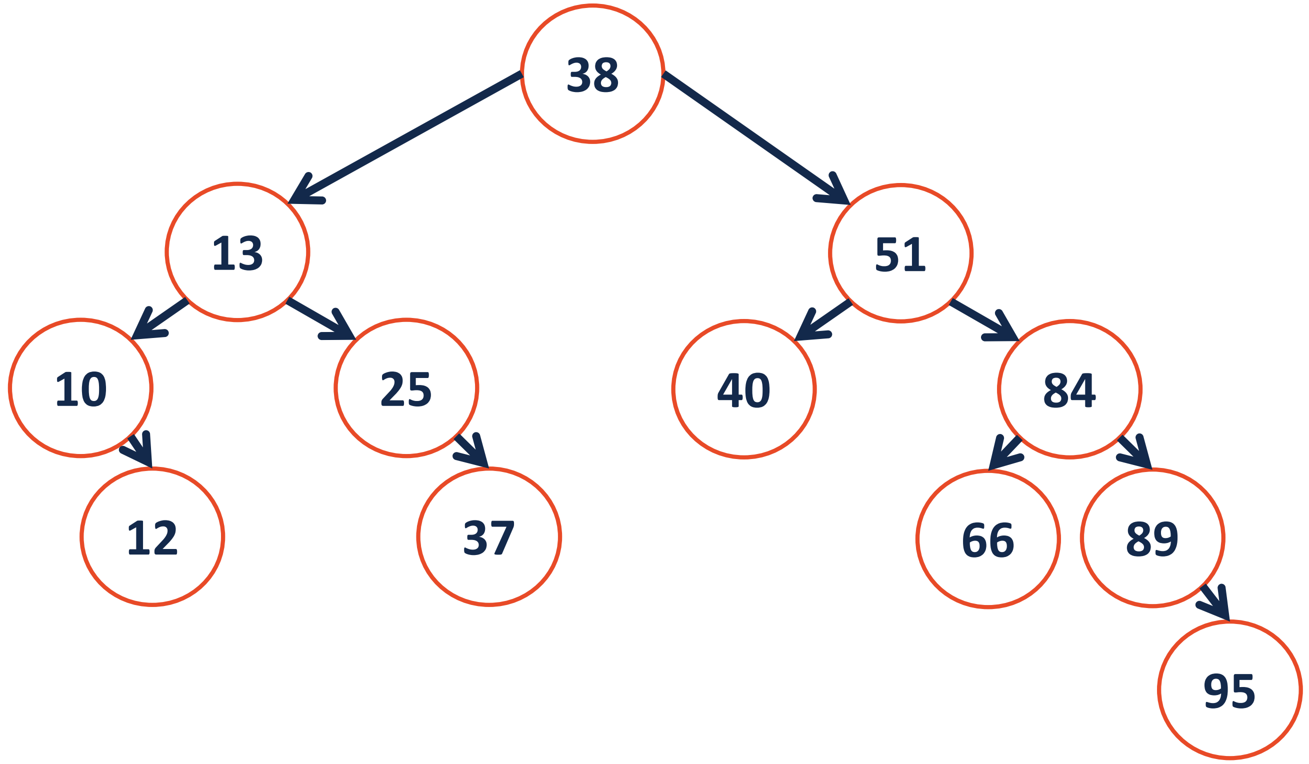
```
segmentation fault
segmentation fault
segmentation fault
```

```
template<typename K, typename V>

_________________________ _remove(TreeNode *& root, const K & key) {
























}
```

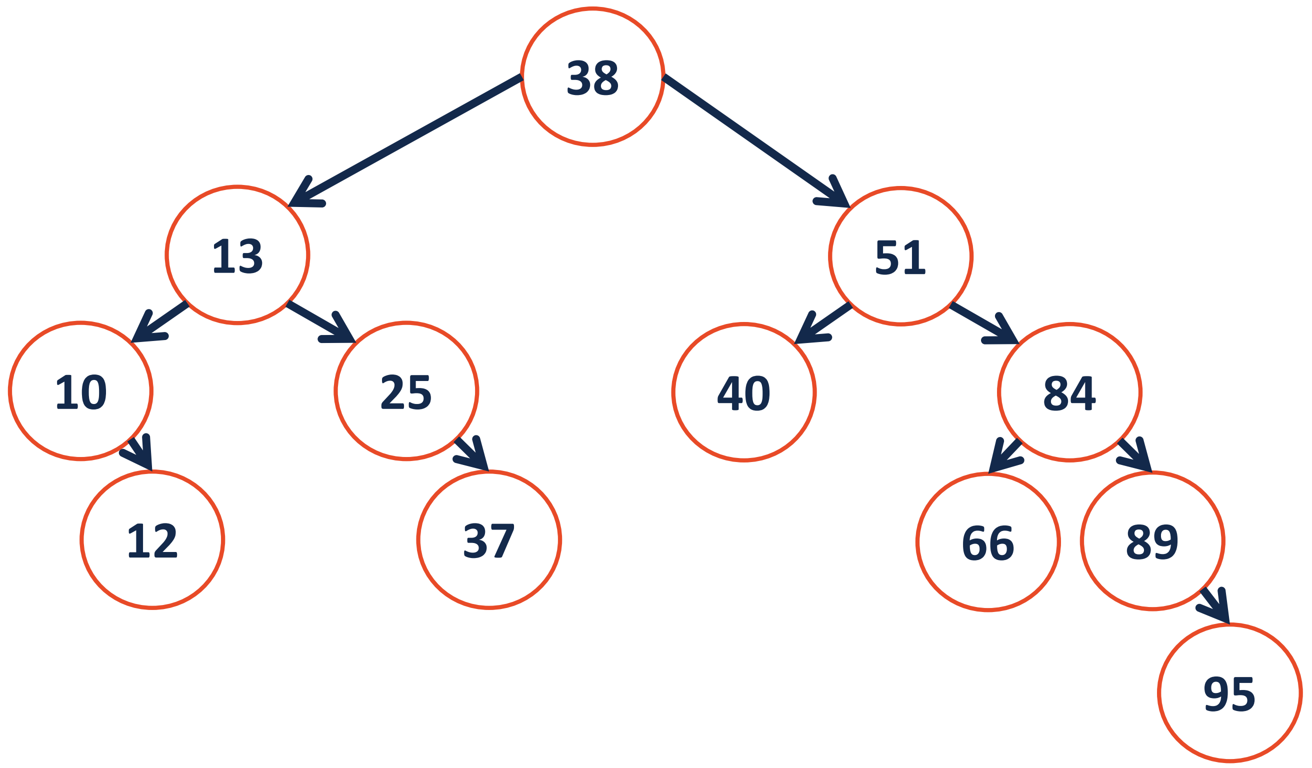




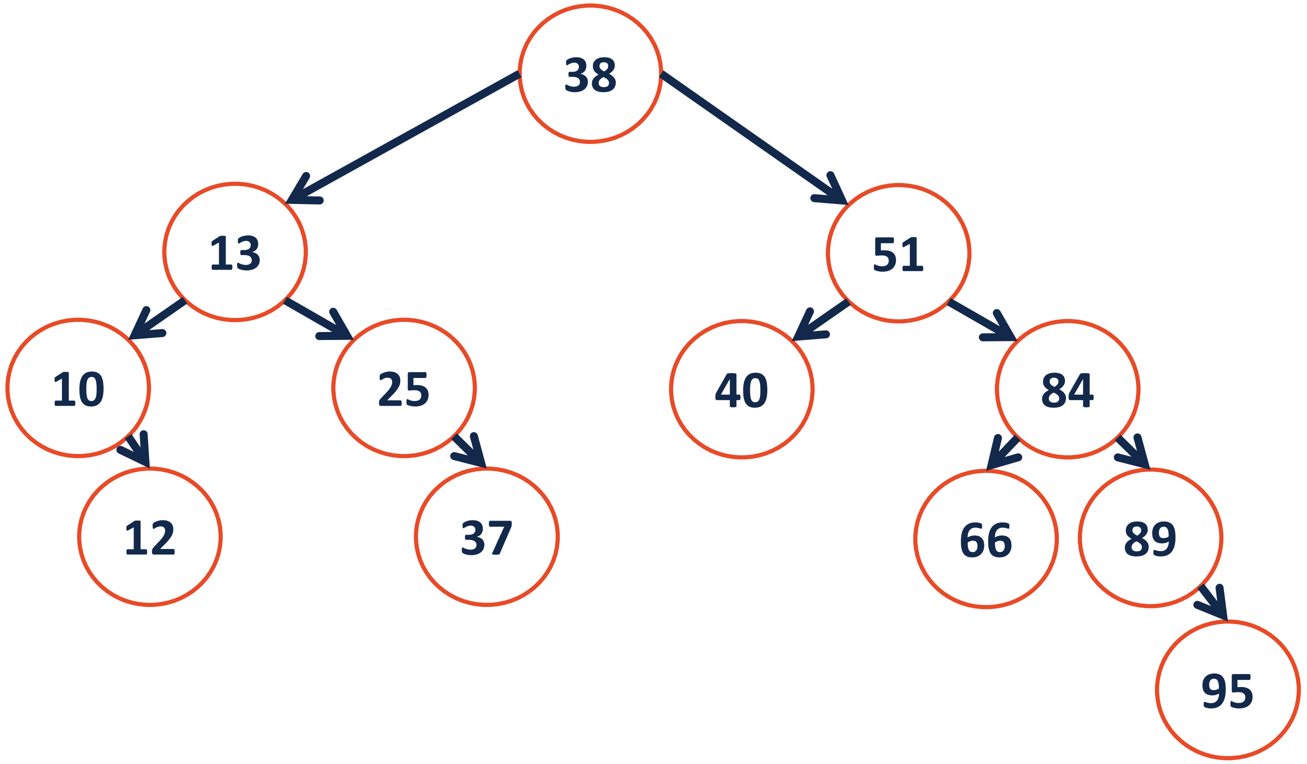
```
remove(40);
```



```
remove(25);
```



```
remove(10);
```



```
remove(13);
```

BST Analysis – Running Time

Operation	BST Worst Case
find	
insert	
delete	
traverse	

BST Analysis

Every operation that we have studied on a BST depends on the height of the tree: **O(h)**.

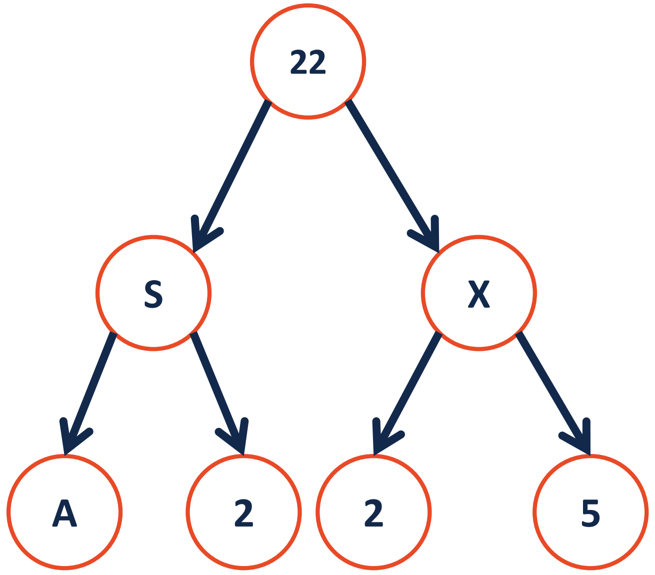
…what is this in terms of **n**, the amount of data?

We need a relationship between h and **n**:

$f(n) \leq h \leq g(n)$

BST Analysis

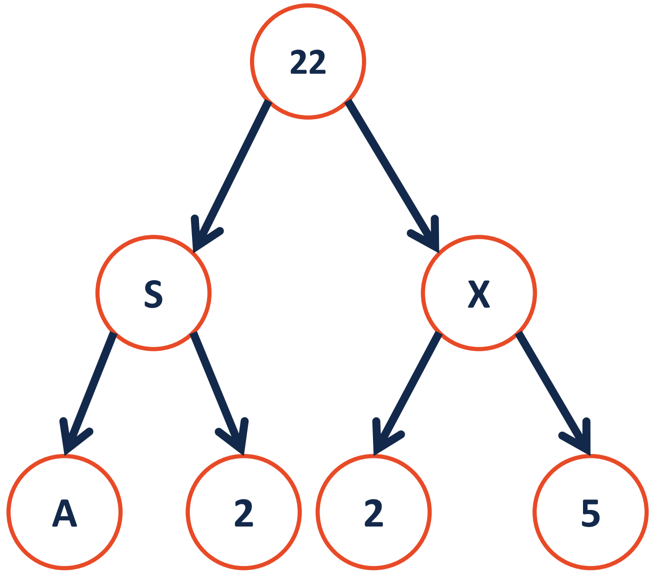
Q: What is the maximum number of nodes in a tree of height **h**?



BST Analysis

Q: What is the minimum number of nodes in a tree of height **h**?

What is the maximum height for a tree of **n** nodes?



BST Analysis

Therefore, for all BST:

Lower bound:

U**pper bound:**

BST Analysis

The height of a BST depends on the order in which the data is inserted into it.

ex: 1 3 2 4 5 7 6 vs. 4 2 3 6 7 1 5

Q: How many different ways are there to insert keys into a BST?

Q: What is the average height of all the arrangements?

BST Analysis

Q: How many different ways are there to insert keys into a BST?

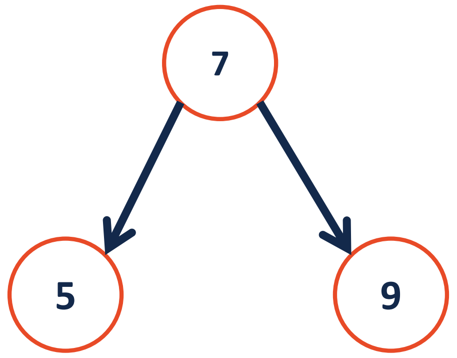
Q: What is the average height of all the arrangements?

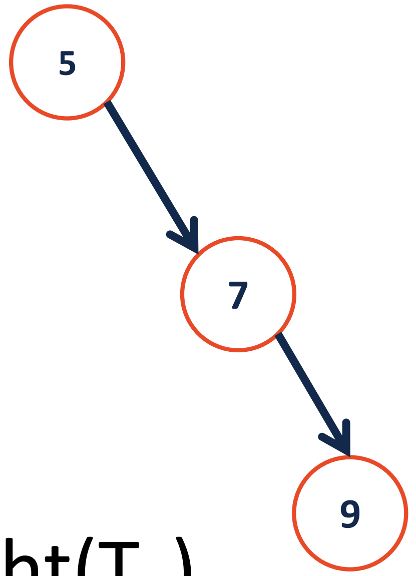
BST Analysis – Running Time

Operation	BST Average case	BST Worst case	Sorted array	Sorted List
find				
insert				
delete				
traverse				

Height-Balanced Tree

What tree makes you happier?





Height balance: $b = \text{height}(T_L) - \text{height}(T_R)$

A tree is height balanced if: