Binary Search Trees for Sentiment Analysis

Laura Burdick - Ph.D. Candidate, University of Michigan

- If you want to follow along and take notes, you can find today's slides at http://laura-burdick.github.io/BST.pdf.

By the end of this lesson, you will be able to:

1. Identify characteristics of a binary search tree (BST).
2. Write pseudo-code for searching a BST.
3. Apply BSTs to the problem of sentiment analysis.

(I will assume prior knowledge of recursion, Big-O notation, and the basics of binary trees.)

Find these slides at http://laura-burdick.github.io/BST.pdf.

Image source: http://lighthouseinsights.in/

Sentiment Analysis

- Deciding if a piece of text conveys positive or negative emotions

A delectable and intriguing thriller filled with surprises, this movie is an original.

A quietly reflective and melancholy New Zealand film about an eventful summer in a 13-year-old girl 's life.

Her film is unrelentingly claustrophobic and unpleasant.

Examples from Stanford Sentiment Treebank V1.0.

Sentiment Analysis

- Deciding if a piece of text conveys positive or negative emotions

It's a bittersweet and lyrical mix of elements.

Although occasionally static to the point of resembling a stage play, the film delivers a solid mixture of sweetness and laughs.

Examples from Stanford Sentiment Treebank V1.0.

Let's Brainstorm!

Given: Movie reviews labeled with sentiment (5-point scale, very positive to very negative) – *training data*

Goal: Classify the sentiment of new reviews that we haven't seen before

What are some ideas to approach this problem?

(this is brainstorming – call out any idea that comes to mind!)

A Simple Approach

- Look at each word that appears in the training data, and calculate its *average sentiment* (the average sentiment of all the sentences that the word appears in).
- Given a new review, take the sentiment of each word in the review (calculated on the training data), and average the sentiment of the words together.

A Simple Approach

Suffers from the lack of a compelling or comprehensible narrative.

-2 0 0 -2 0 0 1 0 0 1

What do we need to implement this?

- A fast way to search for words and find out their average sentiment
- Enter… binary search trees (BSTs)!

Examples from Stanford Sentiment Treebank V1.0.

A Guessing (*Searching*) Game

- Find a partner!
- Person 1: Choose a number between 1 and 15.
- Person 2: Guess the number. Your partner will tell you whether you guessed it correctly, were too high, or too low. Repeat until you guess the number correctly.
- Switch roles, and play again.

This exercise has been modified from Khan Academy (https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/a-guessing-game).

A Guessing (*Searching*) Game

- Discuss with your partner:
  - **What technique did you use to decide which number to guess next?**
- I'll ask several people to share with the group.

This exercise has been modified from Khan Academy (https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/a-guessing-game).

Sequential Search v. Binary Search

Image source: www.mathwarehouse.com

Sequential Search v. Binary Search

Image source: www.mathwarehouse.com

Binary Search Trees (BSTs)

- A **tree** = connected nodes and edges, no cycles, one node designated as the root
- A **binary** tree = each node has at most two children (left child, right child)
- **BST Property** = The left subtree of a node *x* only has nodes that are less than or equal to *x*. The right subtree of a node *x* only has nodes that are greater than or equal to *x*.

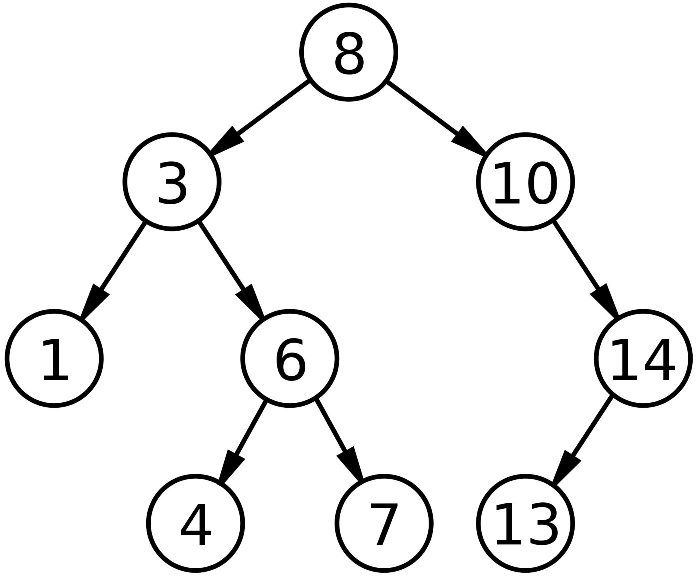


Image source: Wikipedia

"Reading" a BST: Inorder Traversal

```
READ-TREE(x)
    if x ≠ NULL
        then READ-TREE(left[x])
             print x
             READ-TREE(right[x])
```

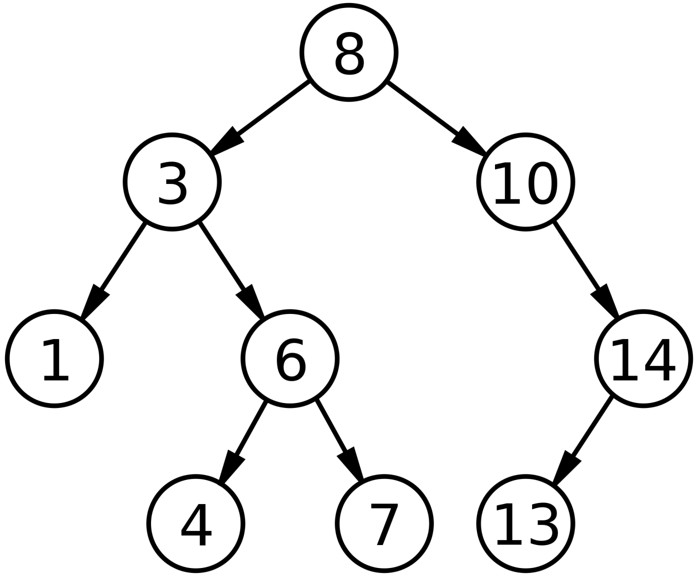


Image source: Wikipedia

Let's Practice!

```
READ-TREE(x)
    if x ≠ NULL
        then READ-TREE(left[x])
             print x
             READ-TREE(right[x])
```

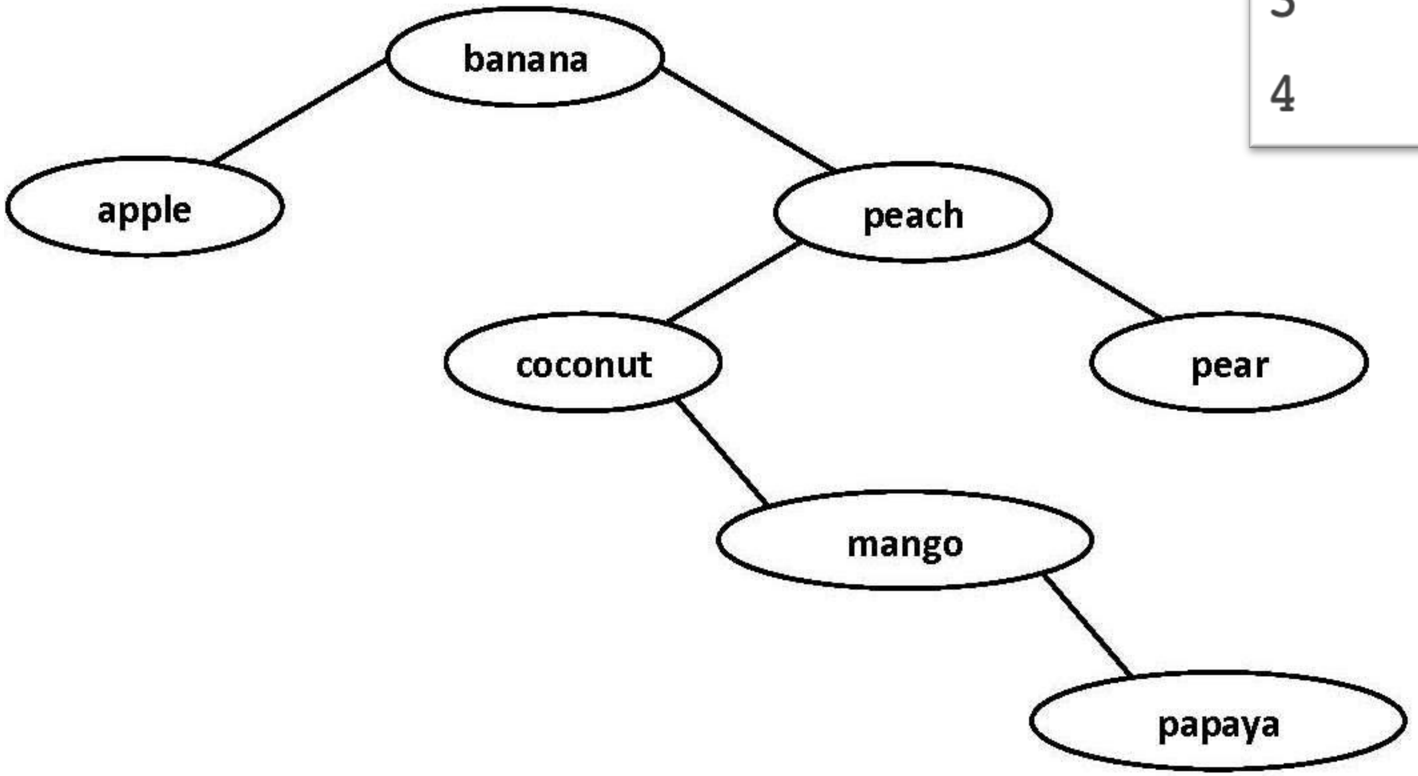


Image source: http://cslearners.blogspot.com/

Searching a BST

- Working individually, write out pseudo-code for the following function:

`TREE-SEARCH(x,v)`

- `x` is the current node you're at (in the beginning, it's the root note)
- `v` is the value you're looking for (e.g., 13)
- *Hint: This will be a recursive function, like the* `READ-TREE(x)` *function!*

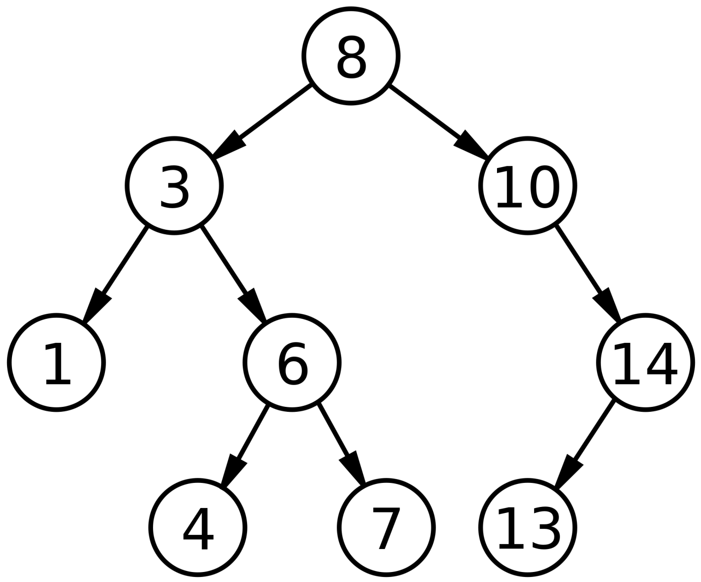
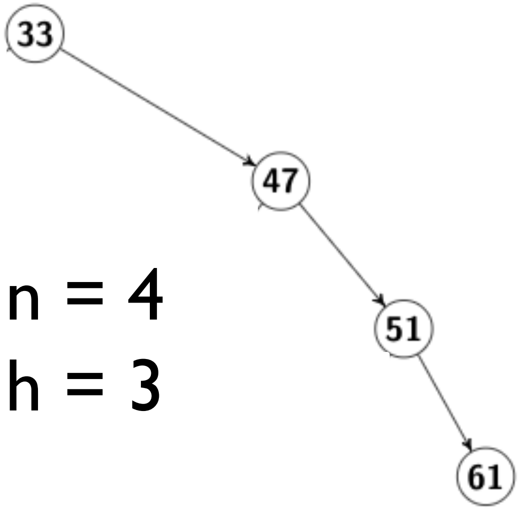


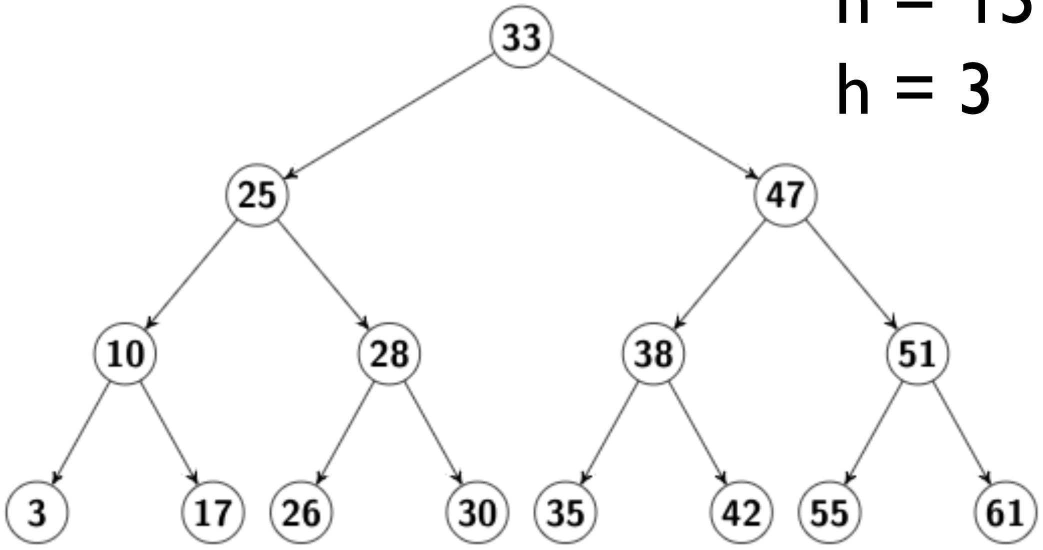
Image source: Wikipedia

How long does it take to find something?

- Searching for an element follows a path downward from the root of the tree. Search is $O(h)$, where h is the height of the tree.
- How tall is a BST?



$n = 4$
$h = 3$

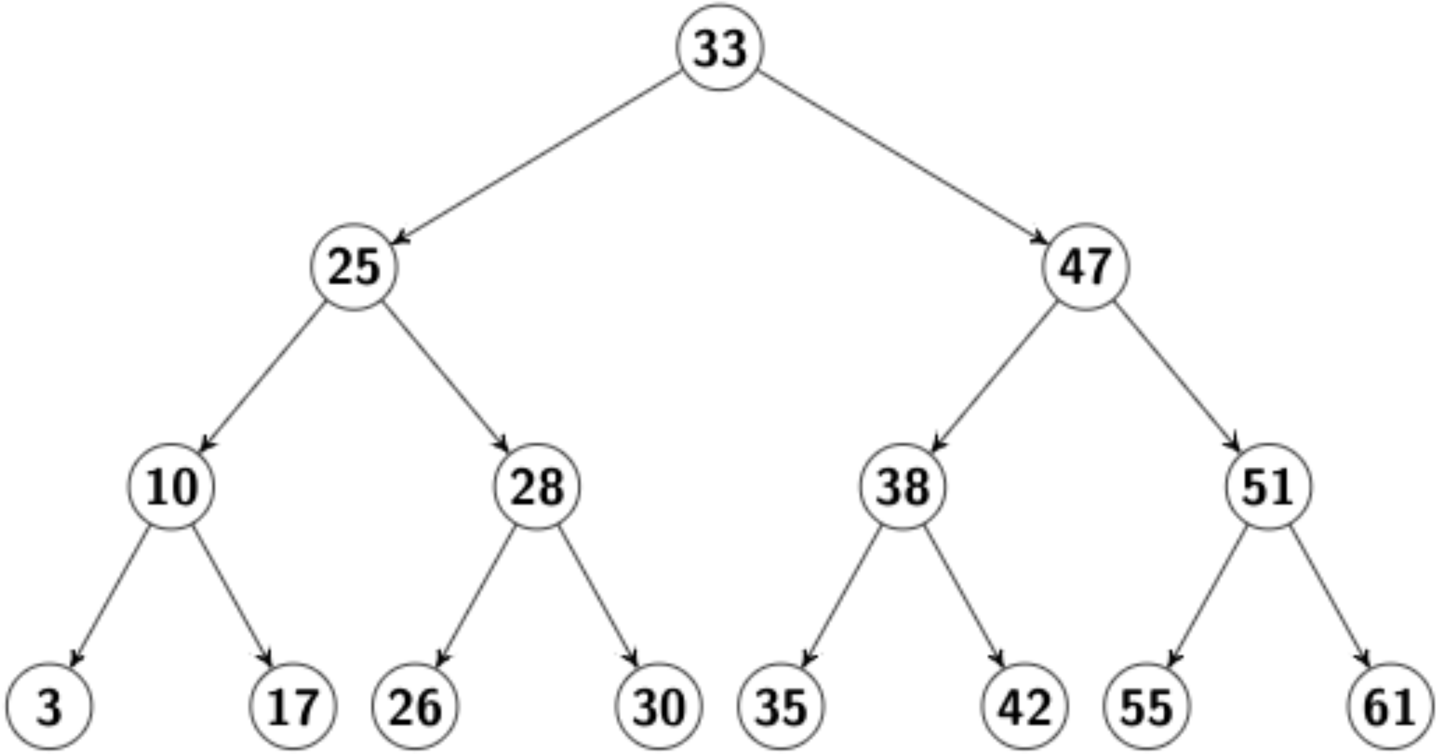


$n = 15$
$h = 3$

How long does it take to find something?

- How tall is a BST?

n = 15
h = lg(n+1)-1
  = *O*(lg(n))
  = 3



1 node

2 nodes

4 nodes

8 nodes

How long does it take to find something?

- Searching for an element follows a path downward from the root of the tree. Search is $O(h)$, where h is the height of the tree.
- Maximum height of a BST is $O(n)$, where n is the number of data points in the tree.
- Minimum height of a BST is $O(lg(n))$.
- We can show for a randomly built BST, expected height is $O(lg(n))$.

1-minute Pause

- Stop and review the material we've covered so far.
- Is there anything that you need clarified?

Image source: http://blackstonevalleyprep.org

Which statement is incorrect?

1. Given a set of numbers, only one valid BST can be constructed containing these numbers.
2. To find the maximum value of a BST, you can follow the right child from the root until a NULL is encountered.
3. To find the minimum value of a BST, you can follow the left child from the root until a NULL is encountered.
4. `READ-TREE(x)` and `TREE-SEARCH(x,v)` can be written in a non-recursive way.

Sentiment Analysis

- Deciding if a piece of text conveys positive or negative emotions

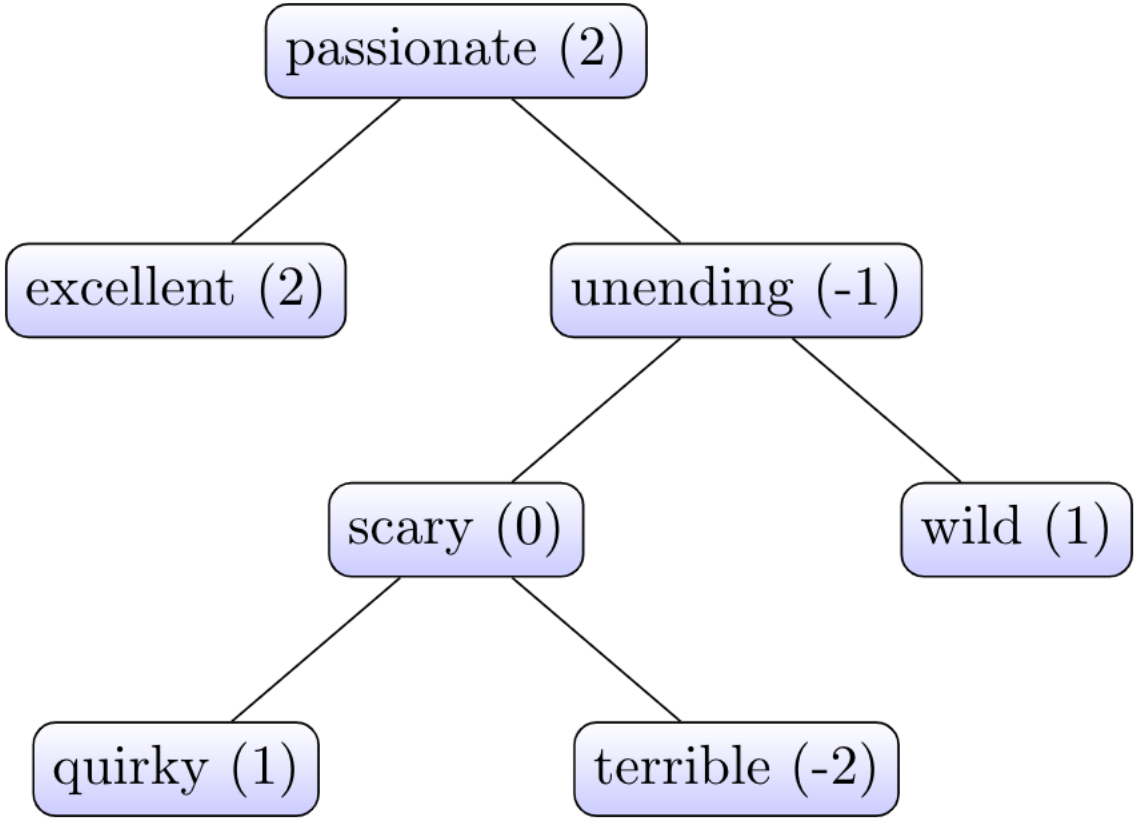
A delectable and intriguing thriller filled with surprises, this movie is an original.

A quietly reflective and melancholy New Zealand film about an eventful summer in a 13-year-old girl 's life .

Her film is unrelentingly claustrophobic and unpleasant.

Examples from Stanford Sentiment Treebank V1.0.

Sentiment Analysis



Main Takeaways

1. Binary search is, on average, more efficient than sequential search.
2. A binary search tree is a data structure that facilitates binary search.
3. Searching through a BST for a value is $O(lg(n))$, where n is the number of data points in the tree.
4. BSTs can be applied to sentiment analysis, determining whether a piece of text is positive or negative.

More Questions?

E-mail me at wenlaura@umich.edu.

Extra Slides!

For your additional enjoyment, or if we have extra time at the end of the lesson.

Inserting into a BST

- `T` is the tree
- `v` is the value to insert (e.g., 20)

`TREE-INSERT(T,v)`

Search for where the new value should go

Insert the new value (rearrange / add tree pointers)

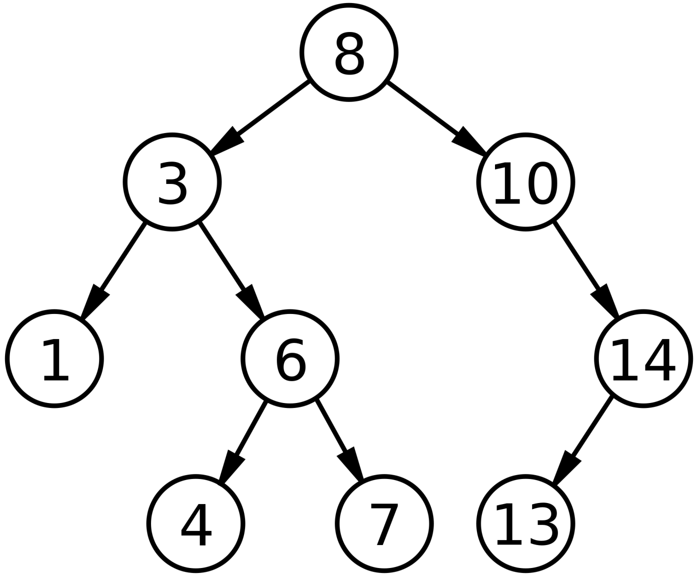


Image source: Wikipedia

Inserting into a BST

- Working individually, write out pseudo-code for the first part of the function (search for where the new value should go).

TREE-INSERT(T,v)

Search for where the new value should go

Insert the new value (rearrange / add tree pointers)

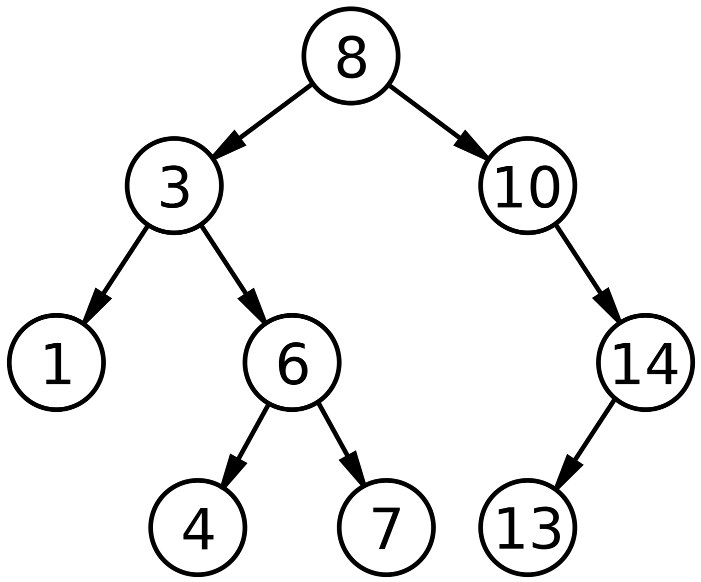


Image source: Wikipedia