CSE 331 Data Structures and Algorithms
Binary Search Trees - BST

Create a BST using incoming numbers, from left to right: [9, 8, 7, 6, 10, 11, 13, 12]
Binary Search Trees - BST

Create a BST using incoming numbers, from left to right.

Steps:

1. 9
2. 9
3. 9
4. 9
5. 9
6. 9
7. 9
8. 9

[9, 8, 7, 6, 10, 11, 13, 12]
AVL Trees 4 Cases for rotations

1. K3 becomes unbalanced on this Insertion

X, Y, Z at height H, K1 is top node of x sub-tree

Y, Z at height H
X is now at height H + 1
K1 is top node of x sub-tree
AVL Trees 4 Cases for rotations

2. X, Y, Z at height $H$, K3 is top node of $z$ subtree

K1 becomes unbalanced on this Insertion

Rebalance

X, Y at height $H$
z is now at height $H + 1$
K3 is top node of $z$ subtree
AVL Trees 4 Cases for rotations

3.

K3 becomes unbalanced on this Insertion
Left sub-tree at height $h + 2$
Right sub-tree (D) at height $h$

A and D at height $h$, B and C at height $h - 1$

A, B, and D at height $h$, C at height $h - 1$
AVL Trees  4 Cases for rotations

K3 becomes unbalanced on this Insertion
Right sub-tree at height $h + 2$
Left sub-tree (A) at height $h$

A, B, and D at height $h$,  
C at height $h - 1$
AVL Trees
Do insertion and then balance

Insert 22 and 20

Left sub tree is empty, Height is -1

Right sub tree has Height 1

Which node became unbalanced?

7 is K3
22 is K1
20 is K2

In this case A, B, C, and D are empty, so their individual heights are all -1.
AVL Trees

Do insertion and then balance

7 is K3
22 is K1
20 is K2

Re-Balance:

Re-Balance:
AVL Trees

Do insertion and then balance

Insert 16

Left sub tree has Height 0

Right sub tree has Height 2

Which node became unbalanced?

6 is K3
20 is K1
7 is K2
Subtree C has one node in it, the node with value 16
Subtree A and D have one node as well
Subtree B has no nodes
AVL Trees

6 is K3
20 is K1
7 is K2
Subtree C has one node in it, the node with value 16
Subtree A and D have one node as well
Subtree B has no nodes
Hashing

Chaining hash functions

Simple Example:

\[ h(x) = x \mod 9 \]
(mod: is modular division, it gives the remainder)
Size \( m = 9 \)

Show the hash table resulting from inserting
In left to right order: \([9, 19, 15, 2, 1, 11, 18]\)

\[
\begin{array}{c|cc}
0 & 9 & 18 \\
1 & 19 & 1 \\
2 & 2 & 11 \\
3 & \\
4 & \\
5 & \\
6 & 15 \\
7 & \\
8 & \\
\end{array}
\]
Hashing

Chaining hash functions

Complex Example: \( h(x) = ((3x + 1) \mod 8) \mod 9 \)

Show the hash table resulting from inserting in left to right order: [9, 19, 15, 2, 1, 11, 18]

Some tips for doing this problem by hand:

When calculating \( x \mod y \), subtract \( y \) from \( x \) until you obtain a value less than \( y \).

\[
\begin{align*}
28 \mod 8 &= 20 \\
20 \mod 8 &= 12 \\
12 \mod 8 &= 4 \\
28 \mod 8 &= 4
\end{align*}
\]

\( h(9) = 4 \mod 9 = 4 \)
BFS and DFS

Show spanning tree resulting from BFS, starting from node v1

Queue Q and Output list O:

<table>
<thead>
<tr>
<th>Process</th>
<th>Q</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFS (V1)</td>
<td>(V1)</td>
<td>()</td>
</tr>
<tr>
<td>Process V1</td>
<td>(V2, V5, V6)</td>
<td>(V1)</td>
</tr>
<tr>
<td>Process V2</td>
<td>(V5, V6)</td>
<td>(V1, V2)</td>
</tr>
<tr>
<td>Process V5</td>
<td>(V6, V7)</td>
<td>(V1, V2, V5)</td>
</tr>
<tr>
<td>Process V6</td>
<td>(V7, V3)</td>
<td>(V1, V2, V5, V6)</td>
</tr>
<tr>
<td>Process V7</td>
<td>(V3)</td>
<td>(V1, V2, V5, V6, V7)</td>
</tr>
<tr>
<td>Process V3</td>
<td>(V4)</td>
<td>(V1, V2, V5, V6, V7, V3)</td>
</tr>
<tr>
<td>Process V4</td>
<td>()</td>
<td>(V1, V2, V5, V6, V7, V3, V4)</td>
</tr>
</tbody>
</table>
BFS and DFS

Show spanning tree resulting from DFS, starting from node V2

Pick the lowest numbered node to follow down
If there are two nodes you can go to, Between V7 and V3 take V3

Stack S and Output list O:

DFS (V2) Process (V2) Process (V3) Process (V4) Process (V1) Process (V6)
S = (V2) S = (V7, V3) S = (V7, V5, V4) S = (V7, V5, V6, V1) S = (V7, V5, V6) S = (V7, V5)
O = () O = (V2) O = (V2, V3) O = (V2, V3, V4) O = (V2, V3, V4, V1) O = (V2, V3, V4, V1, V6)

Process (V5) Process (V7)
S = (V7) S = ()
O = (V2, V3, V4, V1, V6, V5) O = (V2, V3, V4, V1, V6, V5, V7)
Topological Sort

Perform a topological sort on this directed graph. When there are multiple nodes with in-degree zero, choose the lowest numbered vertex to output. For example, if at one step 1, 2, and 3 have zero in-degree, output 1 first.

[ 4 5 7 3 6 8 1 2 10 9 ]