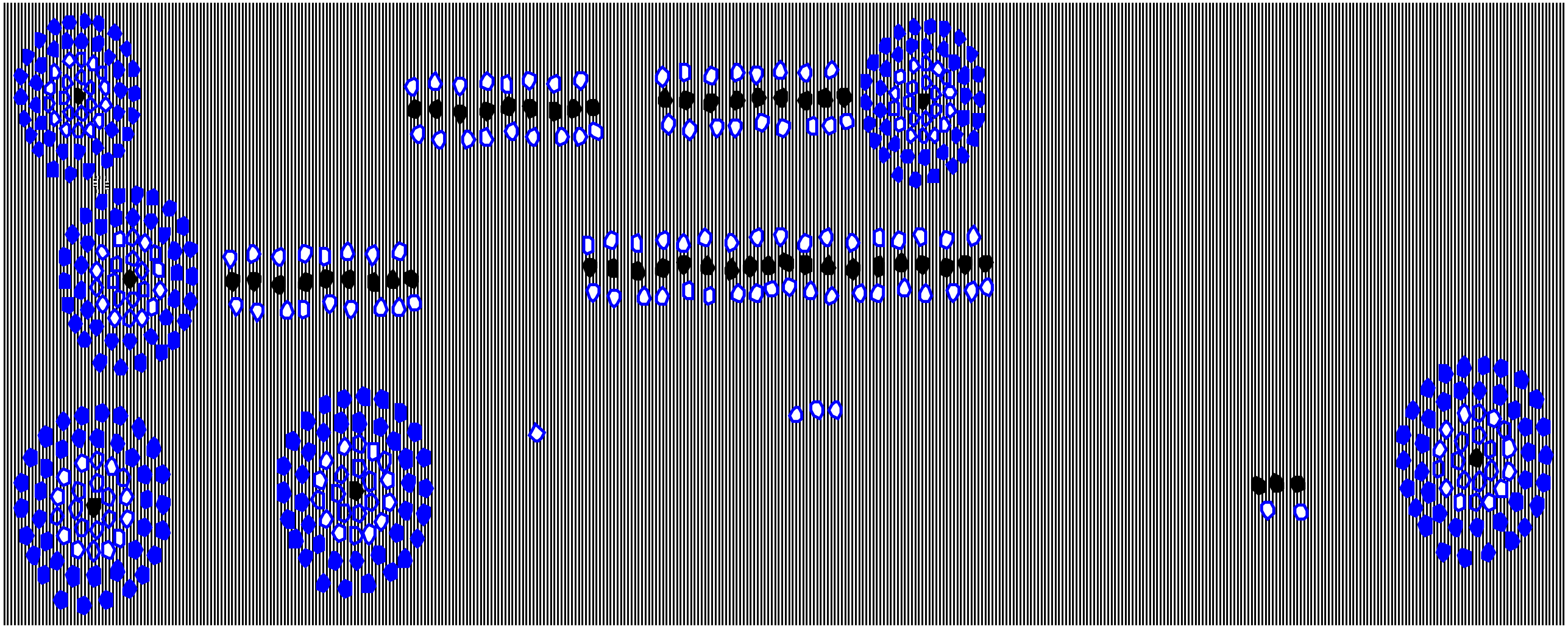
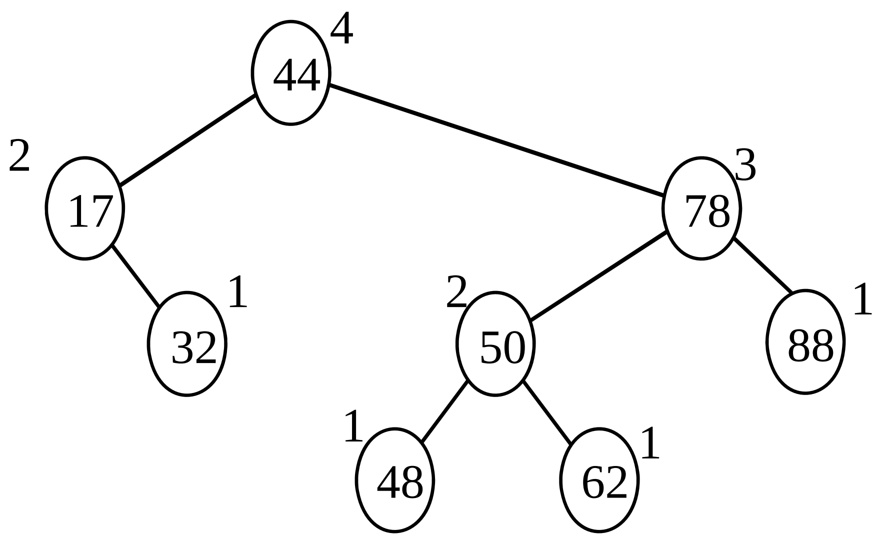
AVL Trees

- AVL Trees



AVL Tree

- **AVL trees are balanced.**
- An AVL Tree is a ***binary search tree*** such that for every internal node v of T, the *heights of the children of v can differ by at most 1*.



An example of an AVL tree where the heights are shown next to the nodes:

Height of an AVL Tree

- **Proposition**: The ***height*** of an AVL tree T storing n keys is O(log n).
- **Justification**: The easiest way to approach this problem is to find **n(h):** the *minimum number of nodes* in an AVL tree of height h.
- We see that n(1) = 1 and n(2) = 2
- For h ≥ 3, an AVL tree of height h contains the root node, one AVL subtree of height h-1 and the other AVL subtree of height h-1 or h-2.
- i.e. n(h) = 1 + n(h-1) + n(h-2)

Height of an AVL Tree (2)

- Knowing $n(h-1) >= n(h-2)$, we get

  $n(h) = n(h-1) + n(h-2) + 1 > 2n(h-2)$

  $n(h) > 2n(h-2)$

  $> 4n(h-4)$

  $> 8n(h-6)$

  …

  $> 2^i n(h-2i)$
- When $i = h/2-1$ we get: $n(h) > 2^{h/2-1} n(2) = 2^{h/2}$
- Taking logarithms: $h < 2\log n(h)$
- Thus the height of an AVL tree is $O(\log n)$

A sharper bound

- We will show how to obtain a sharper bound on the height of an AVL tree.
- We prove using induction that the minimum number of nodes in an AVL tree of height h, $n(h) >= c^h$, where c is some number >1.
- Base case: h=1. Now n(h) >= c > 1.
- Suppose claim is true for all h < k
- We have to show that $n(k) >= c^k$

Sharper bound (2)

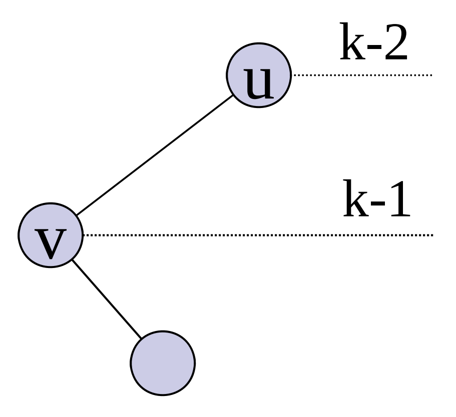
- $n(k) = n(k-1)+n(k-2)+1$
  $>= c^{k-1} + c^{k-2}$ (by induction hypothesis)
- We will be able to show that $n(k) >= c^k$ if we can show that $c^{k-1} + c^{k-2} >= c^k$.
- So c should be such that $c^2-c-1 <= 0$.
- The quadratic equation $c^2-c-1=0$ has roots $\frac{1-\sqrt{5}}{2}$ and $\frac{1+\sqrt{5}}{2}$.
- Hence we can take c as $\frac{1+\sqrt{5}}{2}$ which is roughly 1.63
- Hence AVL tree on n nodes has height atmost $\log_{1.63} n$

Structure of an AVL tree

- Consider an AVL tree on n nodes.
- Consider a leaf which is closest to the root.
- Suppose this leaf is at level k.
- We will show that the height of the tree is at most 2k-1.

Structure of an AVL tree (2)

- Claim: Since closest leaf is at level k all nodes at levels 1..k-2 have 2 children.
- Proof is by contradiction
- Suppose node u at level k-2 has only 1 child, v.
- v is at level k-1 and so cannot be a leaf.
- Hence subtree rooted at v has height at least 2.
- Height-balanced property is violated at u



Structure of an AVL tree (3)

- By previous claim, all levels 1 to k-1 are full.
- Hence tree has at least 2^{k-1} nodes.
- Since height of tree is at most 2k-1 it has at most 2^{2k-1} nodes.
- Thus $2^{k-1} <= n <= 2^{2k-1}$
- Substituting h for 2k-1 we get

$$2^{(h-1)/2} <= n <= 2^h$$

Summary of AVL tree structure

- In an AVL tree of height h, the leaf closest to the root is at level at least (h+1)/2.
- On the first (h-1)/2 levels the AVL tree is a complete binary tree.
- After (h-1)/2 levels the AVL tree may start "thinning out".
- Number of nodes in the AVL tree is at least $2^{(h-1)/2}$ and at most 2^h

Insertion

- A binary tree T is called height-balanced if for every node v, height of v's children differ by atmost one.
- Inserting a node into an AVL tree changes the heights of some of the nodes in T.
- If insertion causes T to become unbalanced, we travel up the tree from the newly created node until we find the first node x such that its grandparent z is unbalanced node.
- Let y be the parent of node x.

Insertion (2)

To rebalance the subtree rooted at z, we must perform a *rotation*.