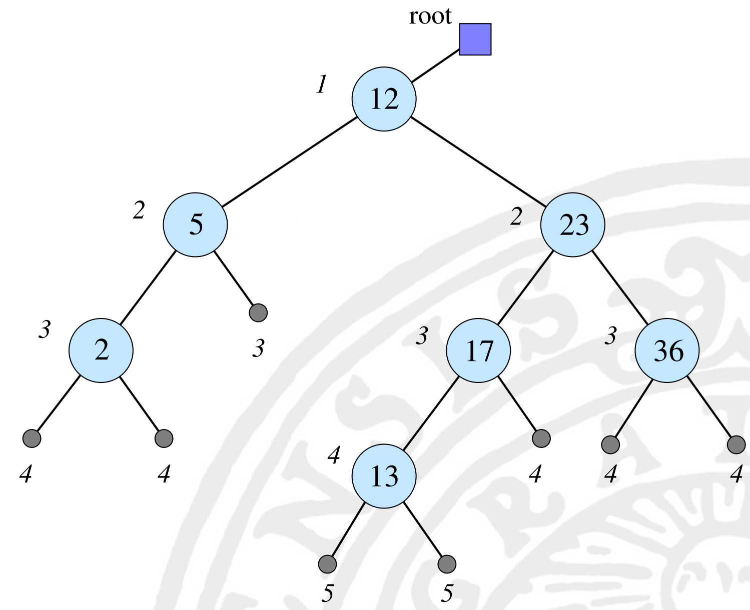
MA3: Properties of binary trees

Tom Smedsaas

This lecture discuss properties of binary trees and their implications for binary search trees.

Measurements on trees

- Size (n): 7
- Height (h): 4
- Internal path length (i or ipl): $1 + 2 + 2 + 3 + 3 + 3 + 4 = 18$
- External path length (e or epl): $4 + 4 + 3 + 5 + 5 + 4 + 4 + 4 = 33$



The two path length properties are a measurements of how well balanced the tree is. They are also linked by the relation $e = i + 2n + 1$

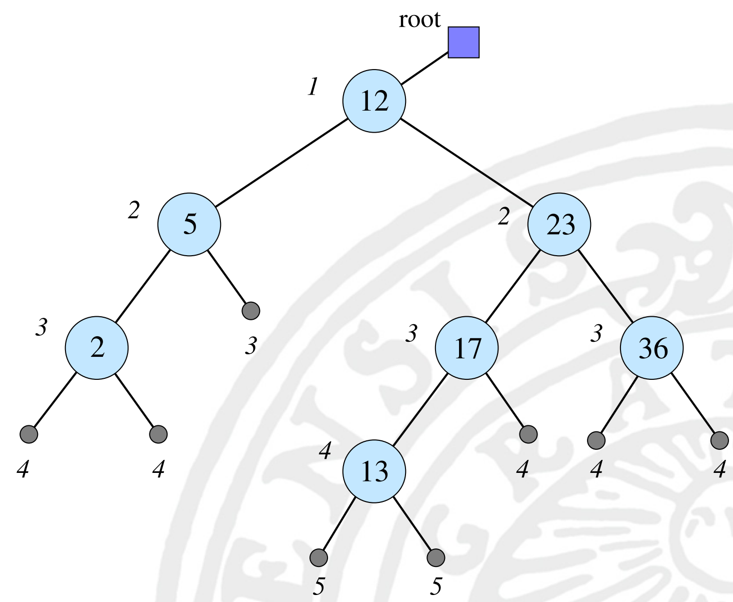
What does these numbers mean?

Searching, *removing* and *inserting* are all O(h) operations.

What about the average?

A *successful* search in a given tree will, *on the average*, require: $\frac{i}{n}$ tries which, in this case, is $\frac{18}{7}$ = 2.57.

An *unsuccessful* search in a given tree will, *on the average*, require: $\frac{e}{n+1}$ tries which, in this case, is $\frac{33}{7+1}$ = 4.12.



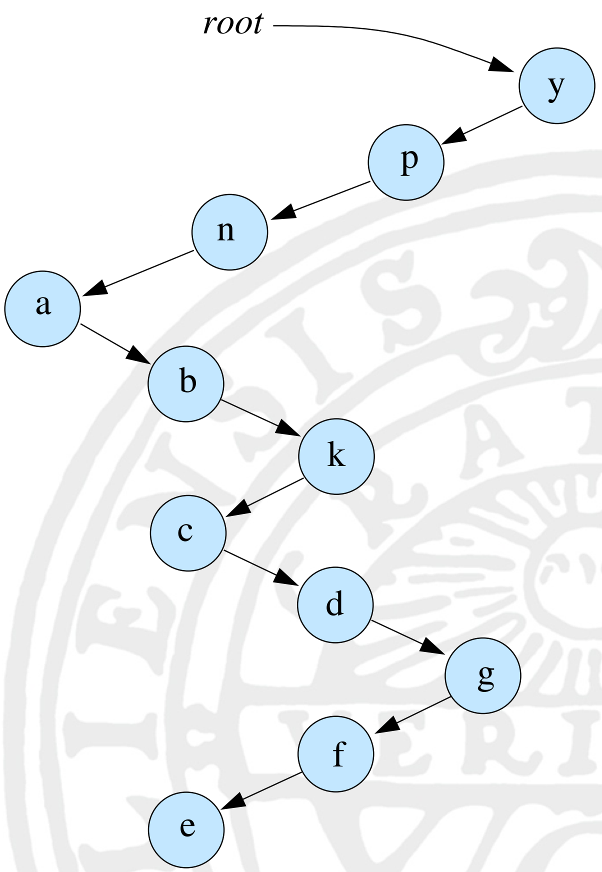
Maximal values?

This tree will give maximal values:

the height is n

$$i = 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

which makes the operations $\Theta(n)$ on the average.

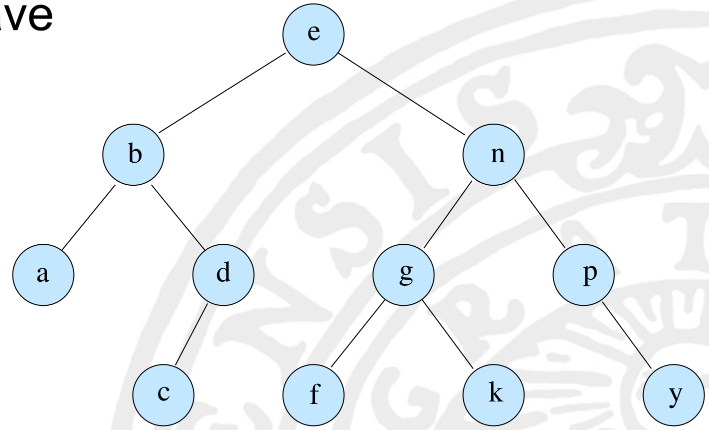


Minimal values

As well branched as possible.

If we have h completely filled levels we will have
$n = 1 + 2 + 4 + \ldots + 2^{h-1} = 2^h - 1$
giving that $h = \log_2(n+1)$

Thus, searching, inserting and and removing keys are all done $O(\log(n))$



Average search tree

What is an average tree?

Suppose we have *n* different keys. There are *n*! possible permutations these keys. If we use each of these permutations to build a tree we can see what the average values of the height and the path lengths are.

It can be shown the the average internal path length over all these trees is

$$1.39 \cdot n \log_2 n + O(n)$$

Thus, for example, searching for a key in a tree with 1000000 keys require, on the average on the average tree $1.39 \cdot \log_2 10^6 \approx 28$ tries.

Thus

The search, insert and remove operations of a key in the ”average” binary search tree with n keys require

$$\frac{1.39n\log_2 n + O(n)}{n} = 1.39\log_2 n + O(1)$$

node visits on the average.

The average case in the worst tree is n node visits. However, there are several insertion algorithms that keep the tree well balanced (AVL-trees, RB-trees …)

The end