

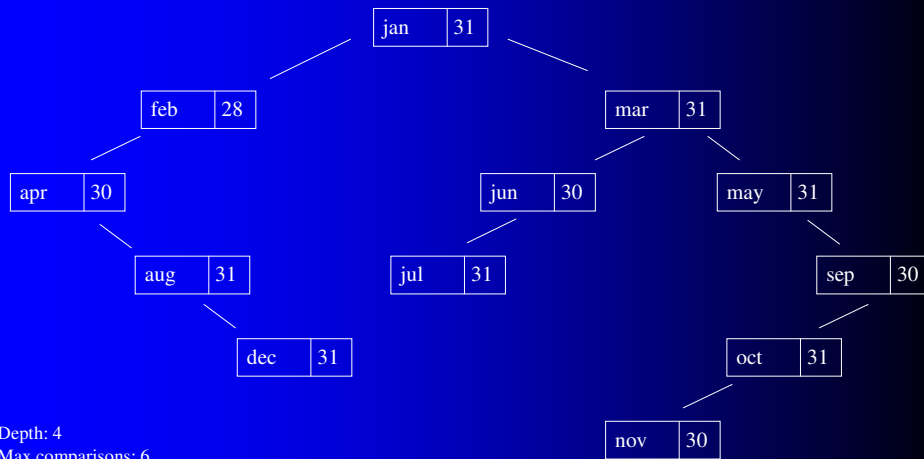
Binary trees

David Morgan

Binary trees

- elements have up to 2 child elements
- left child sorts less, right more, than parent
- tree has a depth
- tree has a balance, comparing depths of its left and right trees (greater difference, less balance)

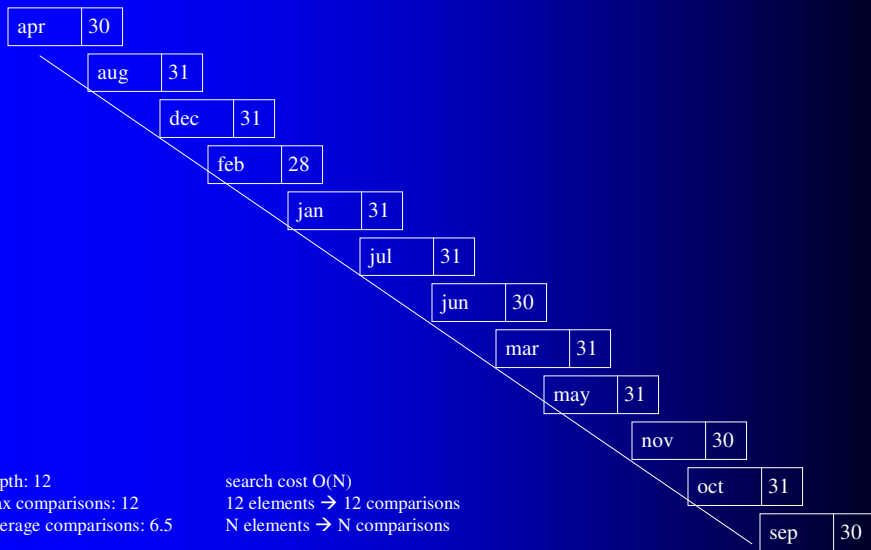
Binary tree of months, for days-per-month determination



Depth: 4
 Max comparisons: 6
 Average comparisons: 3.5

input sequence: jan, feb, mar, apr, may, june, july, aug, sept, oct, nov, dec (chronological)

A skewed tree

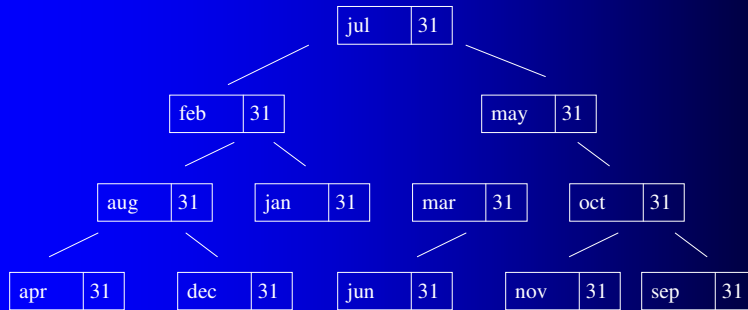


Depth: 12
 Max comparisons: 12
 Average comparisons: 6.5

search cost $O(N)$
 12 elements \rightarrow 12 comparisons
 N elements \rightarrow N comparisons

input sequence: apr, aug, dec, feb, jan, july, june, mar, may, nov, oct, sept (alphabetical)

A balanced tree



search cost $O(\log N)$
 2 levels \rightarrow 3 elements \rightarrow 2 comparisons
 3 levels \rightarrow 7 elements \rightarrow 3 comparisons
 4 levels \rightarrow 15 elements \rightarrow 4 comparisons

Depth: 4

Max comparisons: 4

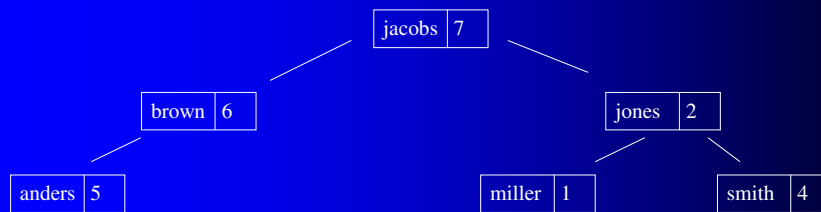
Average comparisons: 3.1

L levels $\rightarrow 2^L - 1$ elements \rightarrow L comparisons, or

$\log(N+1)$ levels \rightarrow N elements $\rightarrow \log(N+1)$ comparisons

input sequence: july, feb, may, aug, dec, mar, oct, apr, jan, june, sept, nov

Binary tree of last names, for data record determination



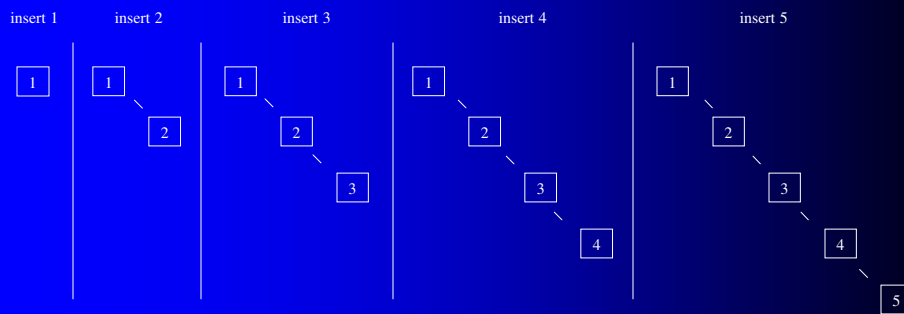
Database

Recno	name	rank	serial_no
1	miller	corporal	4-139
2	jones	major	3-209
3	baker	private	7-981
4	smith	lieutenant	3-101
5	anders	private	8-388
6	brown	sargeant	8-231
7	jacobs	captain	6-495
8	johnson	general	4-556

Tree balance

- depends on insertion sequence
- balance achievable independent of sequence, by performing mid-course re-balancing
 - during insertion, whenever an insertion upsets the balance, re-balance dynamically before inserting next element
 - tree never gets unbalanced, so final result is always balanced

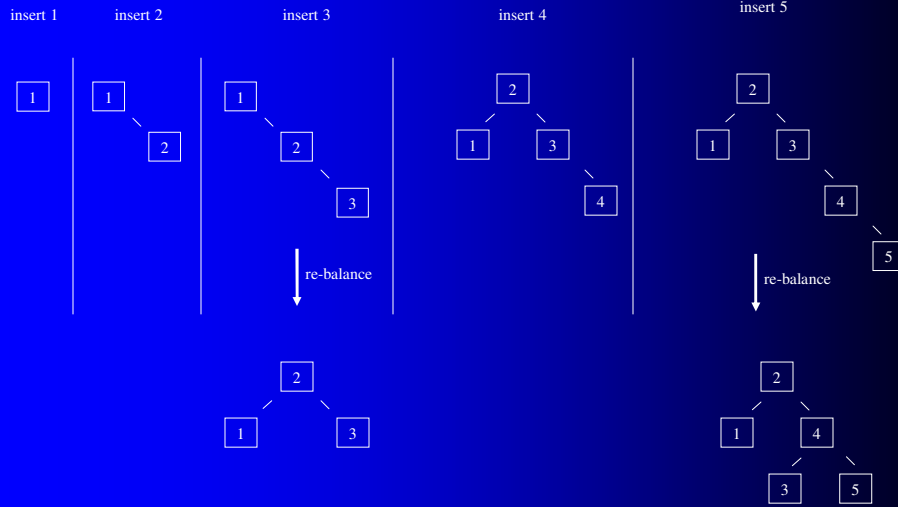
Building tree, no rebalancing



input sequence: 1, 2, 3, 4, 5

final tree unbalanced

Building tree, mid-course re-balancing



input sequence: 1, 2, 3, 4, 5

final tree balanced