

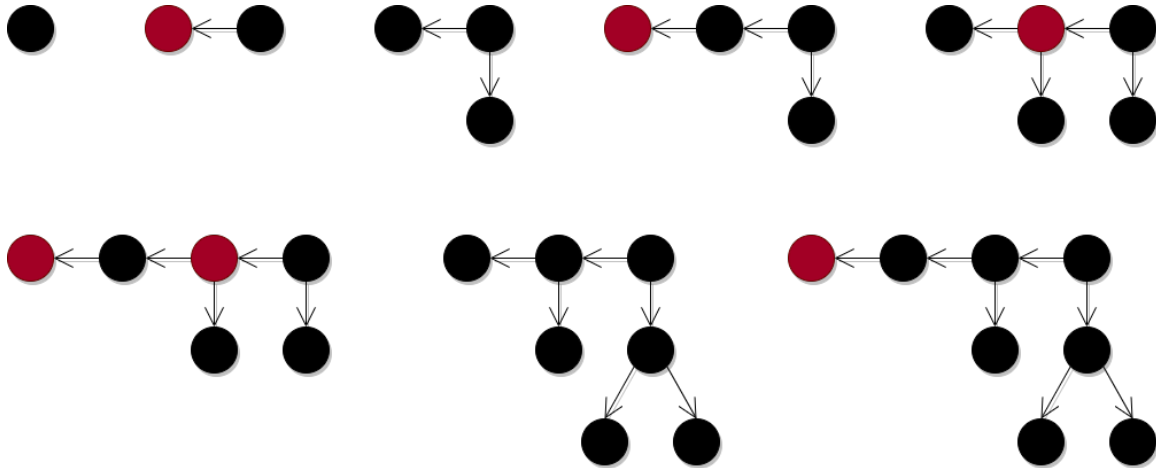
50001 - Algorithm Analysis and Design - Lecture 13

Oliver Killane

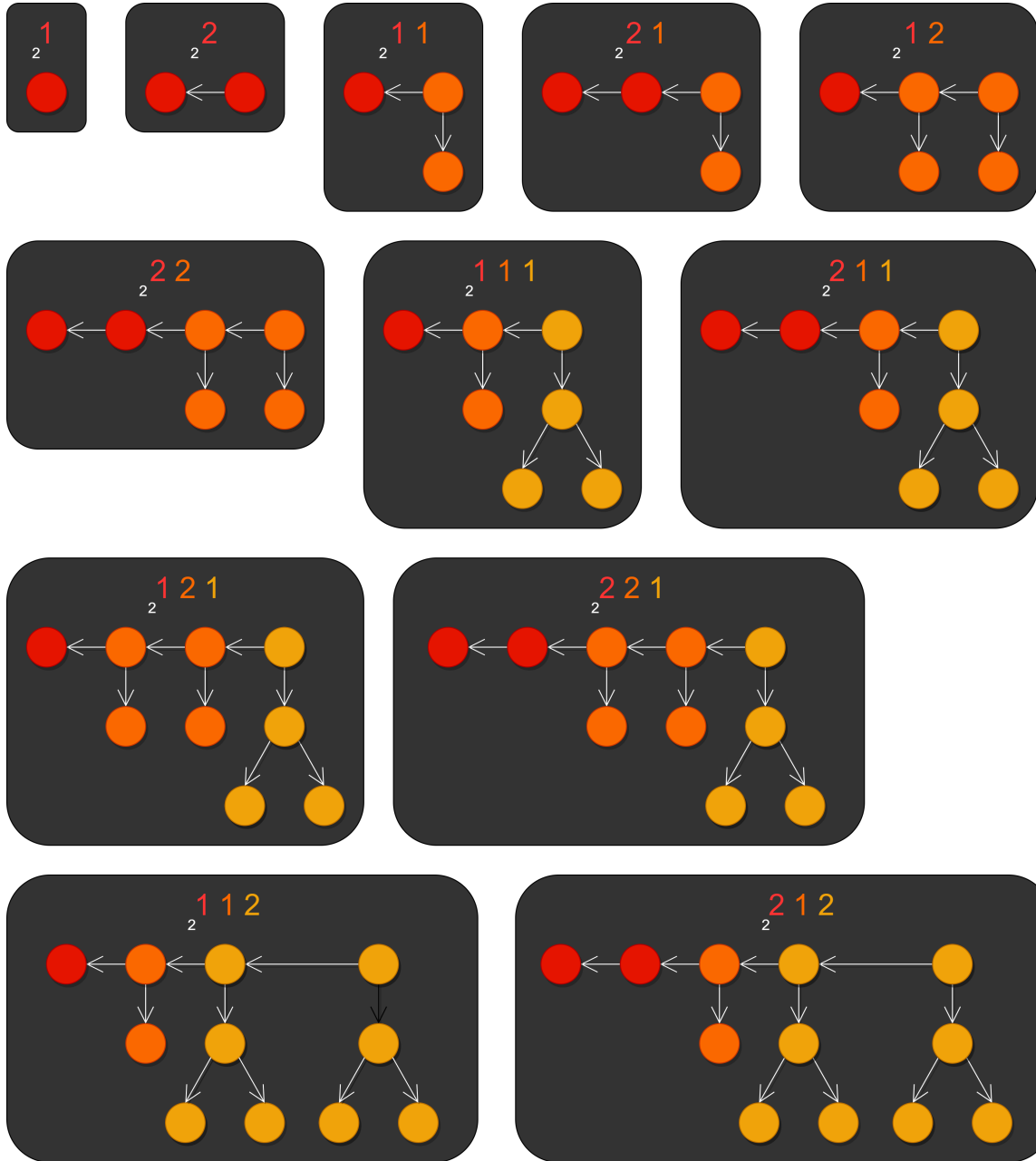
22/11/21

Red Black Trees Continued

We have a pattern with inserting elements from an ordered list into the tree.



We can encode this as a special binary number system, using 1 and 2 such that the least significant bit is the number of trees of 2^0 nodes, and the n th is 2^n .



We can increment this by:

```

1 — each element is a red black tree (+ an extra root element (a))
2 data Digit a = One a (RBTre a) | Two a (RBTre a) a (RBTre a)
3
4 incr :: a -> RBTre a -> [Digit a] -> [Digit a]
5 incr x t [] = [One x t]
6 incr x t ((One y u) : ds) = Two x t y u : ds
7 incr x t ((Two y u z v) : ds) = One x t : incr y (Node Black u z v) ds

```

We can convert a list of digits back to a red black tree by:

```
1  — fold left to combine the digits together into a tree
2  fromList :: [a] -> RBTREE a
3  fromList xs = foldl link Empty (foldr add xs)
4
5  link :: RBTREE a -> Digit a -> RBTREE a
6  link l (One x t) = Node Black l x t
7  link l (Two x t y u) = Node Black (Node Red l x t) y u
```