SWAT

Alphametic Cryptarithms and Polychromatic Coppices

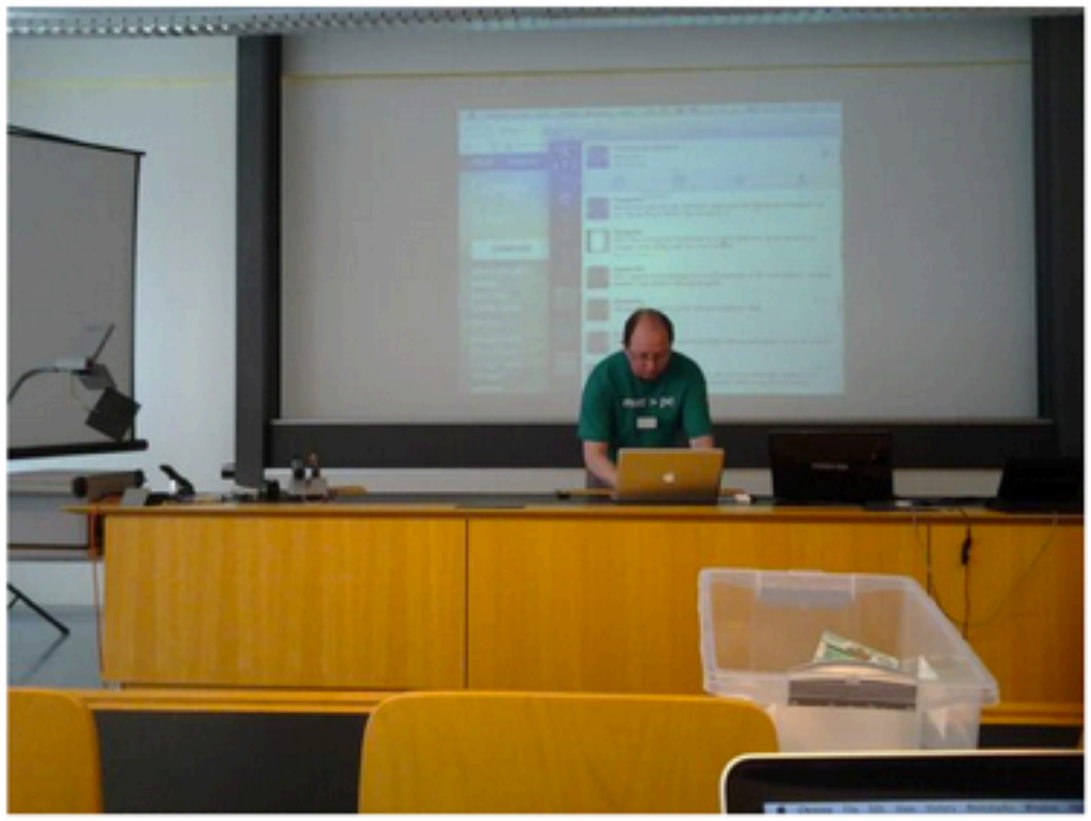
6th SATToSE 2013, Bern

Vadim Zaytsev, SWAT, CWI

Vadim Zaytsev
@grammarware

At #sattose2013, @reallynotabba endeavours to elucidate us the bespeaking of a software chrestomathy.
pic.twitter.com/M43vsAz30N

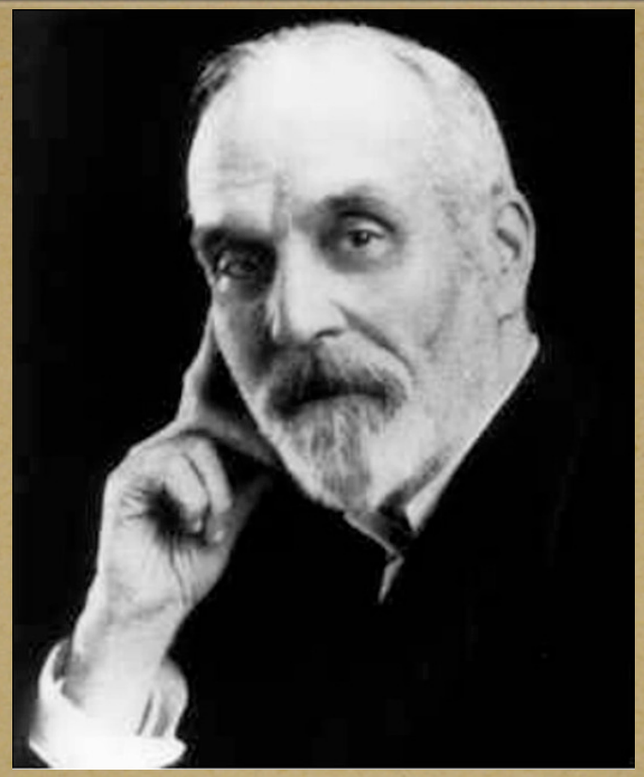
Reply Delete Favorite More



1
RETWEET

11:21 AM - 8 Jul 13 (GMT+02:00) Flag media

Reply to @reallynotabba

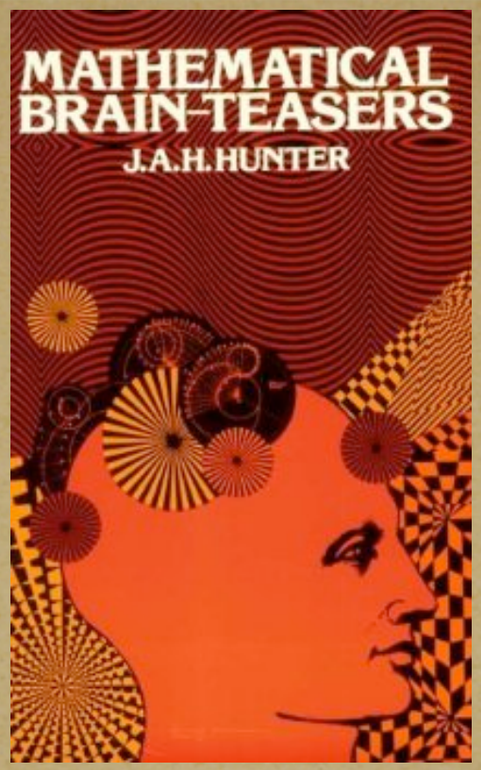


Henry Dudeney, 1924

Cryptarithms

brain teasers obtained when digits in numerical calculations have been replaced by letters

http://www.cut-the-knot.org/cryptarithms/st_crypto.shtml
http://en.wikipedia.org/wiki/File:Henry_Dudeney.jpg



James Hunter, 1955

Alphametics

cryptarithms in which letters form meaningful words, often in meaningful phrases.

http://www.cut-the-knot.org/cryptarithms/st_crypto.shtml
http://www.amazon.com/Mathematical-Brain-Teasers-James-H-Hunter/dp/0486233472

Send more money!

```
  SEND
+ MORE
------
 MONEY
```

A Set-based solution

```
  S E N D
+ M O R E
= M O N E Y
```

```
public set[list[int]] sendMoreMoney(){
   ds = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

   return {[S,E,N,D,M,O,R,Y] |
            int S <- ds,
            int E <- ds - {S},
            int N <- ds - {S, E},
            int D <- ds - {S, E, N},
            int M <- ds - {S, E, N, D},
            int O <- ds - {S, E, N, D, M},
            int R <- ds - {S, E, N, D, M, O},
            int Y <- ds - {S, E, N, D, M, O, R},
            S != 0, M != 0,
                        (S * 1000 + E * 100 + N * 10 + D) +
                        (M * 1000 + O * 100 + R * 10 + E) ==
            (M * 10000 + O * 1000 + N * 100 + E * 10 + Y)};
}
```

```
rascal>sendMoreMoney()
set[list[int]]: {[9,5,6,7,1,0,8,2]}
```

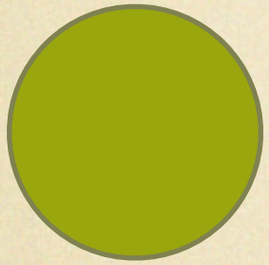
1,814,400 Combinations!

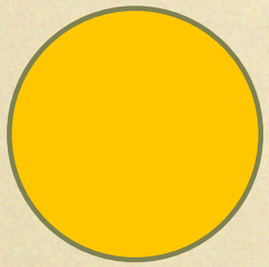
Slides courtesy of Prof. Dr. Paul Klint

Send more money!

```
  9567
+
  1085
------
 10652
```

Green and orange are colours





GREEN

+

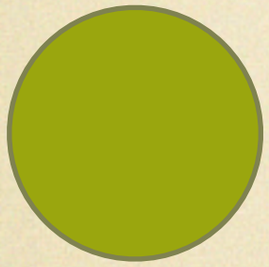
ORANGE

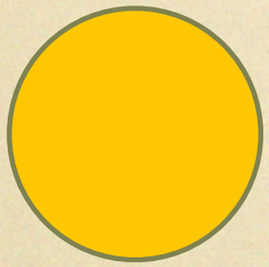
COLORS

Green and orange are colours

```
set[list[int]] colors(){
        ds = {*[0..9]};
        return {[G,R,E,N,O,A,C,L,R,S] |
                int G <- ds,
                int R <- ds - {G},
                int E <- ds - {G,R},
                int N <- ds - {G,R,E},
                int O <- ds - {G,R,E,N},
                int A <- ds - {G,R,E,N,O},
                int C <- ds - {G,R,E,N,O,A},
                int L <- ds - {G,R,E,N,O,A,C},
                int R <- ds - {G,R,E,N,O,A,C,L},
                int S <- ds - {G,R,E,N,O,A,C,L,R},
                G != 0, O != 0, C != 0,
                        (G * 10000 + R * 1000 + E * 100 + E * 10 + N) +
        (O * 100000 + R * 10000 + A * 1000 + N * 100 + G * 10 + E) ==
        (C * 100000 + O * 10000 + L * 1000 + O * 100 + R * 10 + S)};
}
```

Green and orange are colours





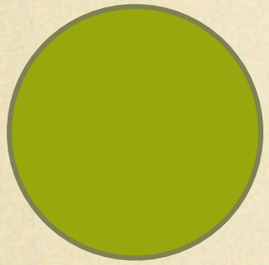
83446

+

135684

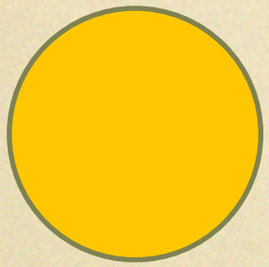
219130

Green and orange are colours



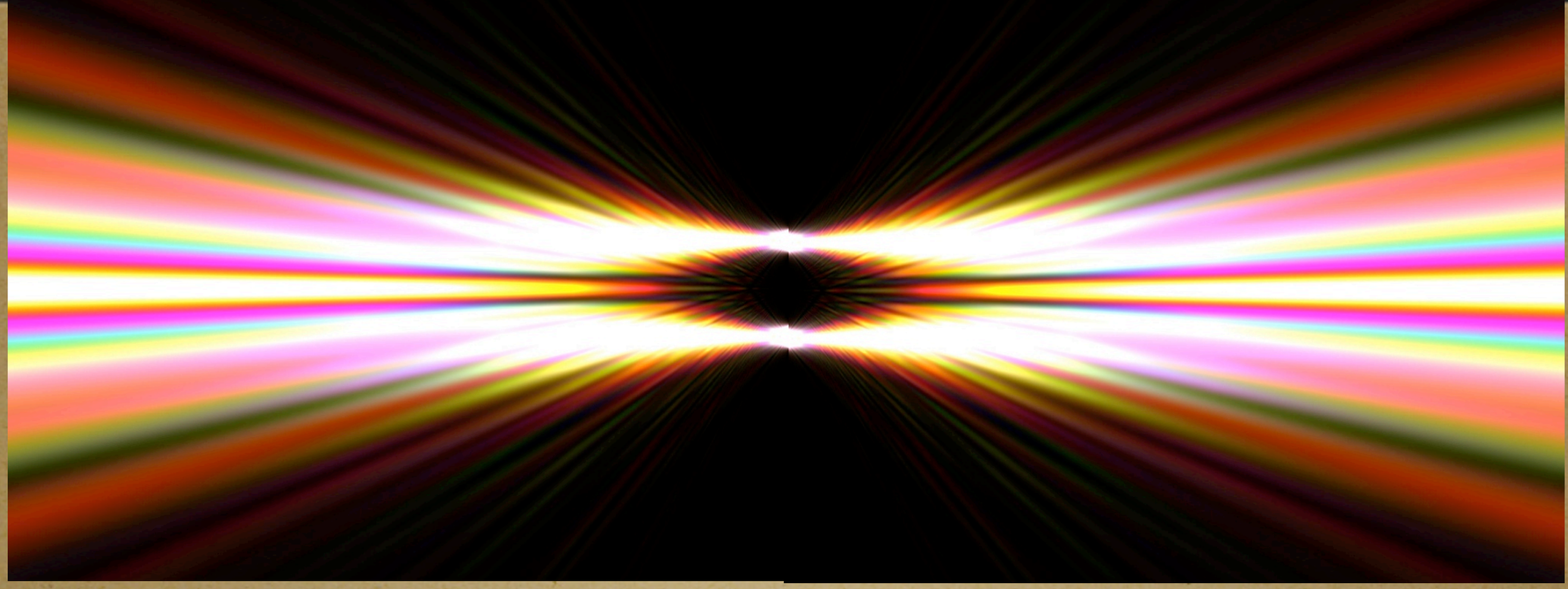
GREEN

\+



ORANGE

COLOURS



Polychromatic

having or exhibiting a variety of colours.

Origin: 1840

http://dictionary.reference.com/browse/Polychromatic
http://commons.wikimedia.org/wiki/File:Double_slit_x-ray_simulation_polychromatic_false-color.jpg

Coppice

a thicket, grove, or growth of small trees

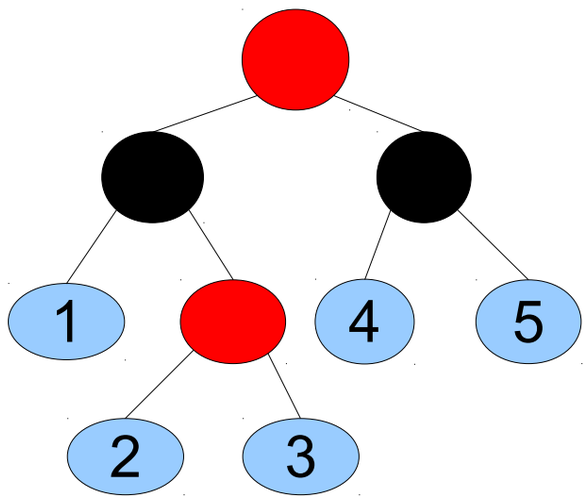
First Known Use: 1534

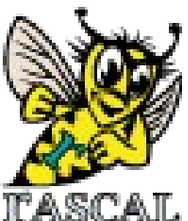
http://www.merriam-webster.com/dictionary/coppice
http://commons.wikimedia.org/wiki/File:Baden%27s_Clump_-_geograph.org.uk_-_489536.jpg

ColoredTrees: *CTree*

```
data CTree = leaf(int N)
           | red(CTree left, CTree right)
           | black(CTree left, CTree right);

public CTree rb =
       red(black(leaf(1), red(leaf(2), leaf(3))),
           black(leaf(4), leaf(5)));
```



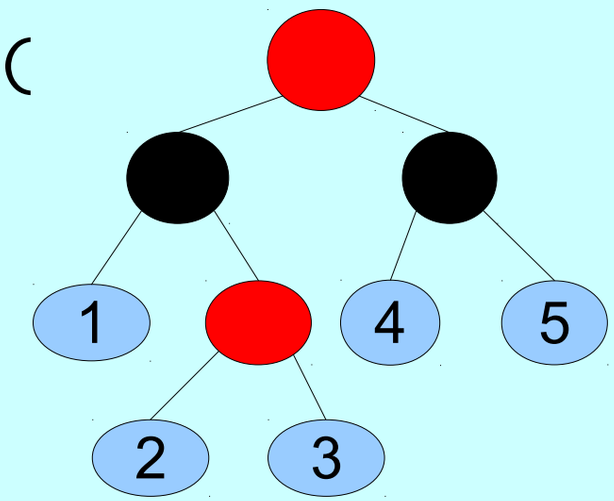


Slides courtesy of Prof. Dr. Paul Klint

Counting Black Nodes, 1

```
public int cntBlack1(CTree t){
    switch(t) {
      case leaf(_):      return 0;
      case black(l,r):   return 1 + cntBlack1(l) + cntBlack1(r);
      case red(l,r):     return cntBlack1(l) + cntBlack1(r);
    };
}
```

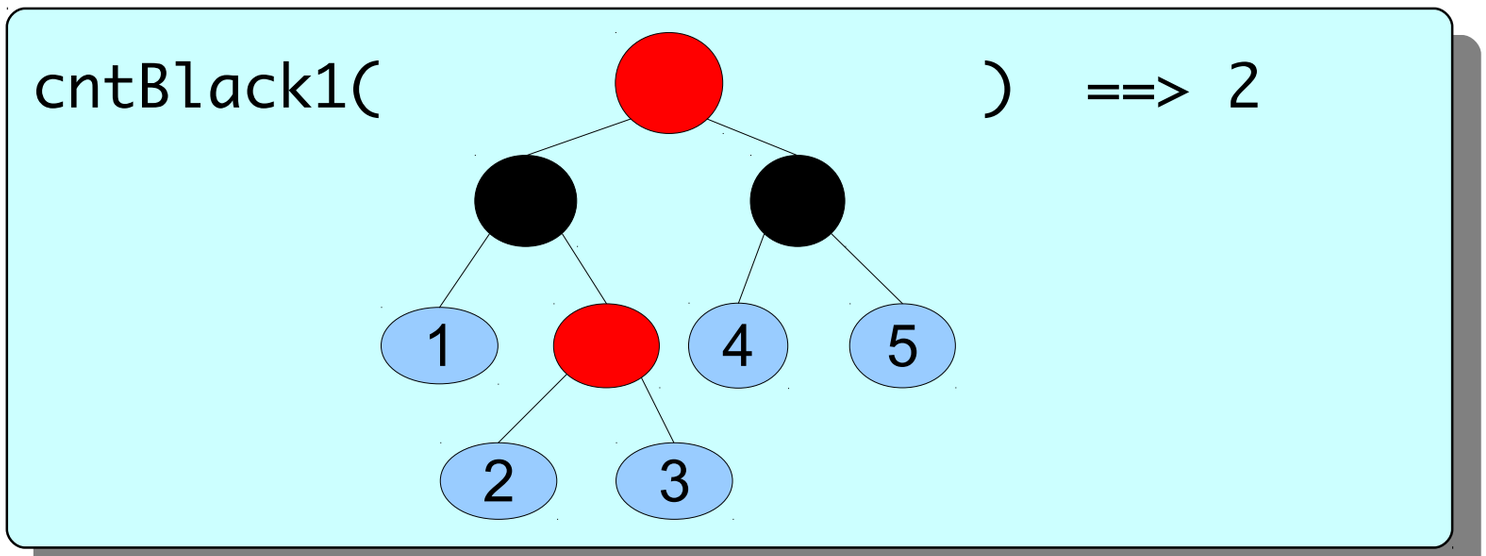
cntBlack1( ) ==> 2



Slides courtesy of Prof. Dr. Paul Klint

Counting Black Nodes, 1

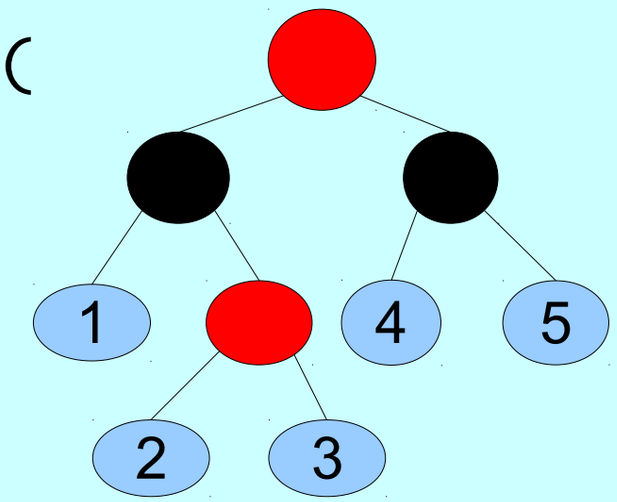
```
public int cntBlack1(CTree t){
    switch(t) {
      case leaf(_):      return 0;
      case black(l,r):   return 1 + cntBlack1(l) + cntBlack1(r);
      case red(l,r):     return cntBlack1(l) + cntBlack1(r);
    };
}
```



Slides courtesy of Prof. Dr. Paul Klint

Counting Black Nodes, 2

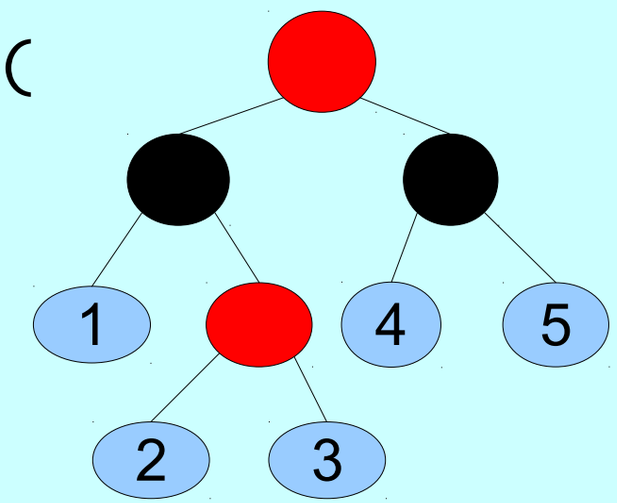
```
public int cntBlack2(CTree t){
    int c = 0;
    visit(t) {
      case black(_,_): c = c + 1;
    };
    return c;
}
```

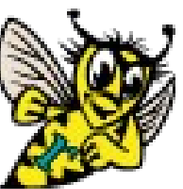
cntBlack2() ==> 2

Slides courtesy of Prof. Dr. Paul Klint

Counting Black Nodes, 2

```
public int cntBlack2(CTree t){
    int c = 0;
    visit(t) {
      case black(_,_): c = c + 1;
    };
    return c;
}
```

cntBlack2() ==> 2



Slides courtesy of Prof. Dr. Paul Klint

```
[n*n | n <- [0 .. 10]]
```

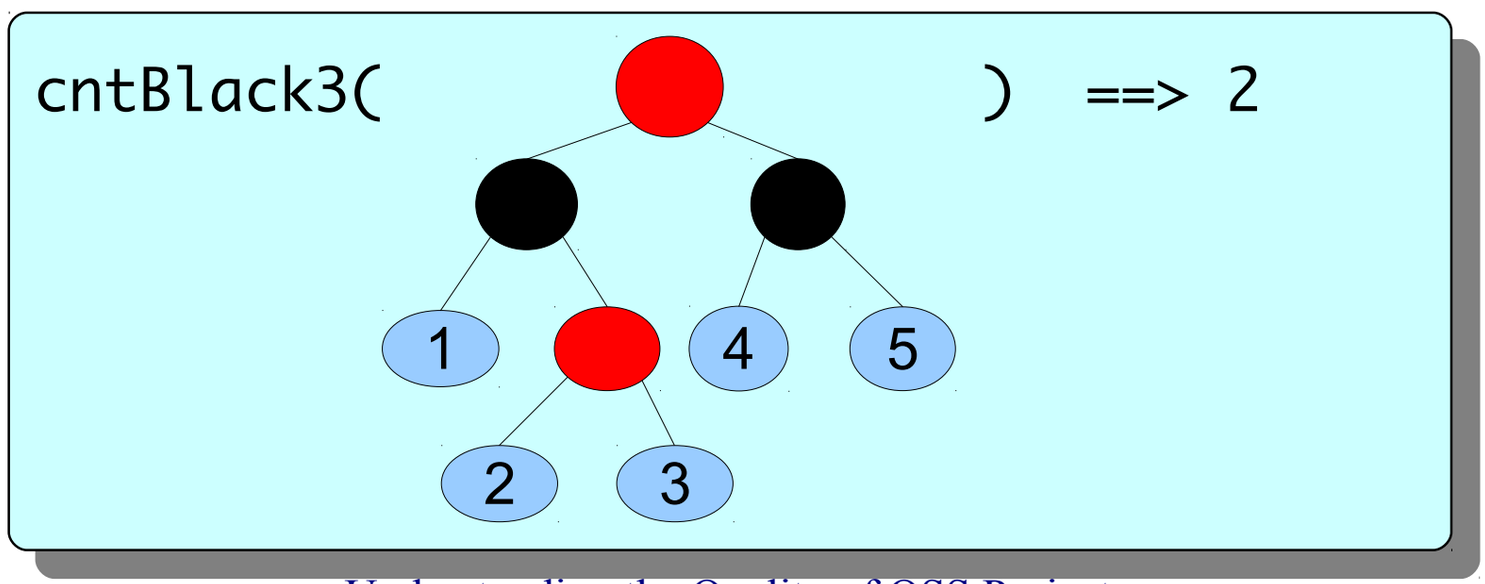
Counting Black Nodes, 3

Deep match of `t` for subtrees `black(_,_)` and bind each to `n`

Pattern match with subject `t`

```
public int cntBlack3(CTree t) = size( [n | /n:black(_,_) := t] );
```

Collect in list comprehension and determine size



To summarise

http://commons.wikimedia.org/wiki/File:Torii_kiyoshige_bando_hikosaburo_ii.jpg

To summarise

- Rascal is a language workbench
- Sets, relations, maps…
- Grammars, ADTs…
- Cool comprehensions
- Cool pattern matching
- Many more features
- http://www.rascal-mpl.org/

http://www.rascal-mpl.org

Questions?

SWAT