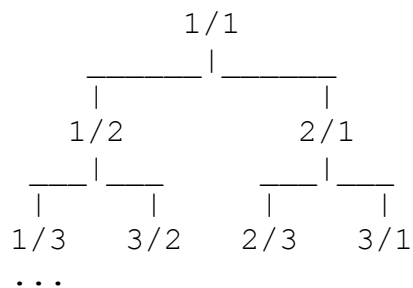


## Kick Start 2013 - Round A

# Analysis: Rational Number Tree



The path to the  $n$ -th element in the tree can be presented by the binary representation of  $n$ . For example, if  $n = 6$ , then its path can be represented by 110, meaning "1 (root,  $1/1$ ) => 1 (go right,  $2/1$ ) => 0 (go left,  $2/3$ )", and we get  $2/3$ . So to solve Q1, we at first find the binary representation of  $n$ , then goes down the tree along the path.

To solve Q2, we need to find the path from  $p/q$  to  $1/1$ . To achieve this we need to continuously subtract  $p$  from  $q$  (if  $q > p$ ) or subtract  $q$  from  $p$  (if  $p > q$ ). Hence, the path can be constructed by representing these two operations by 0 and 1 respectively.

Code list:

```
import fractions
```

```
def Normalize(p, q):
    r = fractions.Fraction(p, q)
    return r.numerator, r.denominator
```

```
def FindElement(n):
    """Solution for Q1."""
    assert n > 0
    path = bin(n)[3:]
    p = q = 1
    for x in path:
        if x == '0':
            q += p
        else:
            p += q
    return p, q
```

```
def FindPosition(p, q):
    """Solution for Q2."""
    assert p > 0 and q > 0
    p, q = Normalize(p, q)
    b = ''
    while p != q:
        if p > q:
            p -= q
            b = '1' + b
        else:
            q -= p
            b = '0' + b
    return b
```

```
    q -= p
    b = '0' + b
b = '1' + b
return int(b, 2)
```