Virahanka Numbers

Abhiram Ranade

Virahanka's Problem (7th century AD?)

Find the number of ways of building an n feet wall using bricks of length 2 feet and 1 foot.

Possibilities for n=8

- 2,2,2,2
- 1,1,1,1,1,1,1,1
- 2,2,2,1,1
- …

Virahanka's Actual Problem

Find number of of poetic meters with 8 beats, made of:

- Short syllables, each 1 beat duration
- Long syllables, each 2 beat duration

Example: *Shardulavikridita* (19 syllables, 15 beats)

L L L S S L S L S S S L L L S L L S L

ya kun den du tu sha r ha r dha va la ya shubh r vas tra vru ta

ya vee na va ra dan d man di ta k ra ya shwe t pad ma s na

Aa ji chya ja wa lii gha dya l ka sa lay aa he ch mat kaa ri ka

De ii the vu ni te ku the a ju n hii na hi ku na tha u ka

Trial and error solution

P_i = Set of all i beat patterns

$V_i = |P_i|$ = number of patterns having i beats.

$V_1 = 1 : P_1 = \{S\}$

$V_2 = 2 : P_2 = \{SS, L\}$

$V_3 = 3 : P_3 = \{SL, SSS, LS\}$

$V_4 = 5 : P_4 = \{SSL, LL, SLS, SSSS, LSS\}$

Virahanka's Solution

- "By the method of Pingala, it is enough to observe that the last syllable is long or short"
- Pingala: mathematician/poet from 300 B.C.
- Virahanka is giving credit to someone who lived 1000 years before him!!
- Copy but give credit..

Virahanka's solution

P_i = set of i beat patterns

L_i = set of i beat patterns ending in L

S_i = set of i beat patterns ending in S

$P_i = L_i \cup S_i$ … disjoint union

$|P_i| = |L_i| + |S_i|$

$|L_i|$

Let p = any pattern in L_8

f(p) = p with last syllable, L, removed.

Thus f(p) is in P_6

Let p’ be any pattern in P_6. f^{-1}(p) exists? Yes, = p’L

f is 1-1

Thus $|L_8| = |P_6|$ $|L_i| = |P_{i-2}|$

Similarly $|S_8| = |P_7|$ $|S_i| = |P_{i-1}|$

$|P_i|$

$|P_i| = |L_i| + |S_i|$

$= |P_{i-2}| + |P_{i-1}|$

$V_i = V_{i-2} + V_{i-1}$

$V_1 = 1$, $V_2 = 2$, found by exhaustive trial and error

$V_3 = V_1 + V_2 = 1 + 2 = 3$

$V_4 = V_2 + V_3 = 2 + 3 = 5$

…

Recursive function

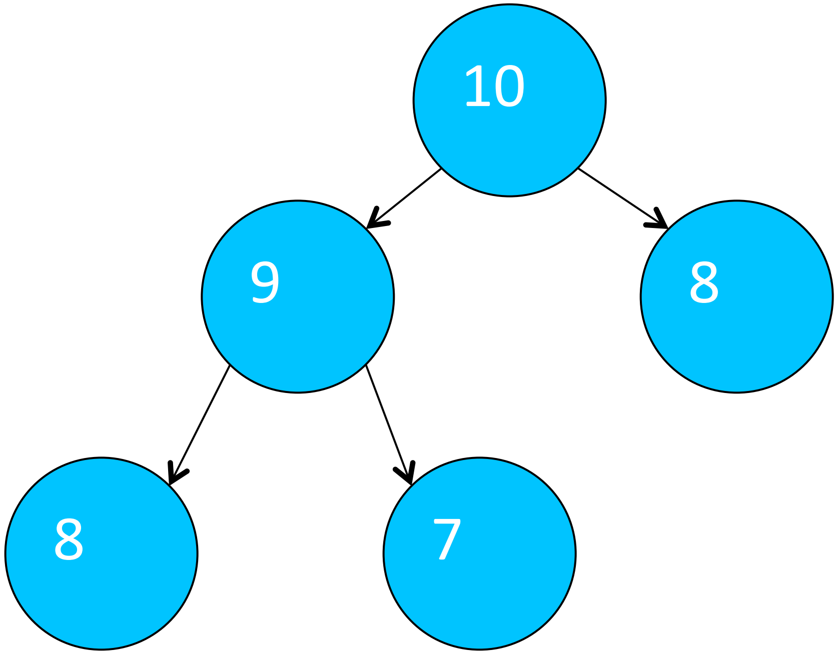
```
int Virahanka(int n){
  if(n == 1) return 1;
  if(n == 2) return 2;
  return Virahanka(n-1) + Virahanka(n-2);
}

int main(){ cout << Virahanka(10) << endl;
```

Iterative Program

```
int n; cin >> n;  // Need nth Virahanka number
int v1 = 1, v2 = 2;
int vlast = v2, vsecondlast = v1;
repeat(n-2){
  int vcur = vlast + vsecondlast;
  vsecondlast = vlast;            // This iteration's last =
                                  // next iterations' secondlast
  vlast = vcur;
}
cout << vlast << endl;
```

Which is better?

"Recursion tree for recursive program"



Which is better (contd.)

Recursive function: computation is repeated.

Time proportional to $2^{n/2}$.

Not practical even for small values.

Summary

Recursion: way of discovering algorithms.

- Relate solution of large problems to solution of small problems of same type.
- Extremely powerful

Recursion: way of writing programs.

- Elegant, but sometimes may be inefficient

On Virahanka Numbers

- Series is very interesting.
  - Number of petals in many flowers.
  - Ratio of consecutive terms tends to a limit.
- Mathematics from poetry!
- More commonly known as Fibonacci numbers, though Virahanka lived well before Fibonacci.
- What has gone wrong with our country in the last 1000 years? What do we need to do?

Remark on functions

- Functions that return values may be called without using the returned value. e.g.

```
getClick();
```

- Useful when function has a side effect and output.