The Fibonacci Sequence and Memoization

The Fibonacci sequence was introduced to Western Europe by Leonardo Bonacci of Pisa, also known as Fibonacci, who lived from 1170 to around 1240, possibly the greatest mathematician of Medieval Europe. The Fibonacci numbers are defined recursively:

$F_0 = 0$
$F_1 = 1$
$F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.

The first few Fibonacci numbers are given in the following table.

n	F_n
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34

Computing the Fibonacci Numbers

The simplest way to compute the first N Fibonacci numbers is by dynamic programming:

```
F[0] = 0;
F[1] = 1;
for n = 2toN
  F[n] = F[n-1]+F[n-2];
```

There is lesser knwon recurrence which generates the Fibonacci sequence.

$F_0 = 0$
$F_1 = 1$
$F_2 = 1$
$F_n = F_{(n-1)/2} * F_{n/2} + F_{(n+1)/2} * F_{(n+2)/2}$ for $n > 2$

Here is an alternative expression of the same recurrence:
$F_0 = 0$
$F_1 = 1$
$F_2 = 1$

$$F_n = \begin{cases} F^2_{(n-1)/2} + F^2_{(n+1)/2} & \text{if } n \text{ odd and greater than 1} \\ F_{n/2-1} * F_{n/2} + F_{n/2} * F_{n/2+1} & \text{if } n \text{ even and greater than 2} \end{cases}$$

For example, $F_7 = F_3^2 + F_4^2 = 4 + 9 = 13$ and $F_8 = F_3 * F_4 + F_4 * F_5 = 6 + 15 = 21$..

Computation of F_N using memoization and the second recurrence requires $\Theta(\log N)$ time, The idea is that the search structure needs to store only $\Theta(\log N)$ memos, specifically, memos of the form (n, F_n) for values of n close to $\frac{N}{2^k}$ for $k \leq \log_2 N$. The following figure illustrates that set of values for $N = 1000$. An arrow from i to j in the figure means that F_j is needed to compute F_i.