Fibonacci Numbers and Recurrences

Lecture 23
CS2110 – Fall 2015

Fibonacci (Leonardo Pisano) 1170-1240?
Statue in Pisa Italy

Prelim 2 Thursday at 5:30 and 7:30

5:30. Statler Auditorium. Last name M-Z
7:30. Statler Auditorium. Last name A-L
If you completed assignment P2Conflict and did not hear from us, just go to your desired time
26 people 7:30 -> 5:30
07 people 5:30 -> 7:30
If you are AUTHORIZED BY CORNELL to have a quiet place or extend time and completed P2Conflict, go to Gates 405, starting from 5:15.

About implementing heaps

/* Add e with priority p to the priority queue.
 * Throw an IllegalArgumentException if e is already in the queue.
 * Expected time is O(log N), worst-case time is O(N).
 */
public void add(E e, double p) {
    if (b.contains(e)) throw new …);  
    b.add(size,e);  size= size+1;  
    int ni= b.indexOf(e);  size= size-1;  
}

Operations size, isEmpty, get, set, iterator, and listIterator run in constant time. Operation add runs in amortized constant time, that is, adding n elements requires O(n) time. All the other operations run in linear time (roughly speaking). The constant factor is low compared to that for the LinkedList implementation.

Creating unnecessary objects

In bubble-up

b.set(k, child);
map.put(child, new Info(k, p_child));

BETTER:

b.set(k, child);
map.get(child).index= k;

More effective presentation

1. Put { on line before, with space before
2. Put space after if, else, while do, etc.
3. If the “then-part” of an if statement returns or throws, do not use else.

public E peek() {
    if(b.size() == 0) {
        throw new PC…();
    } else {
        return b.get(0);
    }
}
Fibonacci function

```java
/** Return fib(n). Precondition: n ≥ 0. */
public static int f(int n) {
    if (n <= 1) return n;
    return f(n-1) + f(n-2);
}
```

Fibonacci in nature

The artichoke uses the Fibonacci pattern to spiral the sprouts of its flowers.

The artichoke sprouts its leaves at a constant amount of rotation: 222.5 degrees (in other words the distance between one leaf and the next is 222.5 degrees). You can measure this rotation by dividing 360 degrees (a full spin) by the inverse of the golden ratio. We see later how the golden ratio is connected to fibonacci.

topones.weebly.com/1/post/2012/10/the-artichoke-and-fibonacci.html

Fibonacci in Pascal’s Triangle

\[ p(i,j) \] is the number of ways \( i \) elements can be chosen from a set of size \( j \)
Blooms: strobe-animated sculptures

www.instructables.com/id/Blooming-Zoetrope-Sculptures/

Uses of Fibonacci sequence in CS

Fibonacci search
Fibonacci heap data structure
Fibonacci cubes: graphs used for interconnecting parallel and distributed systems

Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2)
\end{align*}
\]

We can prove that \( T(n) \) is \( O(2^n) \)

It's a "proof by induction".
Proof by induction is not covered in this course.
But we can give you an idea about why \( T(n) \) is \( O(2^n) \)

\( T(n) \leq c \cdot 2^n \) for \( n \geq N \)

Reorder for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2)
\end{align*}
\]

\( T(n) \leq c \cdot 2^n \) for \( n \geq N \)

Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2)
\end{align*}
\]

\( T(n) \leq c \cdot 2^n \) for \( n \geq N \)

Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2)
\end{align*}
\]

\( T(n) \leq c \cdot 2^n \) for \( n \geq N \)

Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2)
\end{align*}
\]

\( T(n) \leq c \cdot 2^n \) for \( n \geq N \)
Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2) \\
T(0) &= a \leq a \cdot 2^0 \\
T(1) &= a \leq a \cdot 2^1 \\
T(2) &= a \leq a \cdot 2^2 \\
T(3) &= a \leq a \cdot 2^3 \\
T(4) &= a \leq a \cdot 2^4
\end{align*}
\]

Recursion for fib: \( f(n) = f(n-1) + f(n-2) \)

\[
egin{align*}
T(0) &= a \\
T(1) &= a \\
T(n) &= T(n-1) + T(n-2) \\
T(0) &= a \leq a \cdot 2^0 \\
T(1) &= a \leq a \cdot 2^1 \\
T(2) &= a \leq a \cdot 2^2 \\
T(3) &= a \leq a \cdot 2^3 \\
T(4) &= a \leq a \cdot 2^4
\end{align*}
\]

The golden ratio

\( \frac{a + b}{a} = \frac{a}{b} \) call that value \( \psi \)

\[
\psi^2 = \psi + 1 \quad \text{so} \quad \psi = \frac{1 + \sqrt{5}}{2} = 1.618 \ldots
\]

1.618…

ratio of sum of sides to longer side

\[
\frac{a}{b} = \text{ratio of longer side to shorter side}
\]

The golden ratio

How to draw a golden rectangle
The Parthenon

Can prove that Fibonacci recurrence is $O(\phi^n)$

We won’t prove it.
Requires proof by induction
Relies on identity $\phi^2 = \phi + 1$

Linear algorithm to calculate $\text{fib}(n)$

```java
/** Return fib(n), for n >= 0. */
public static int f(int n) {
    if (n <= 1) return 1;
    int p = 0;  int c = 1;  int i = 2;
    // invariant: p = fib(i-2) and c = fib(i-1)
    while (i < n) {
        int fibi = c + p;   p = c;  c = fibi;
        i = i+1;
    }
    return c + p;
}
```

Logarithmic algorithm!

```
f_0 = 0
f_1 = 1
f_{n+2} = f_{n+1} + f_n
```

You know a logarithmic algorithm for exponentiation — recursive and iterative versions

\[
\begin{pmatrix}
0 & 1 \\
1 & 1 \\
\end{pmatrix}^n
= \begin{pmatrix}
f_n \\
f_{n+1}
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 \\
1 & 1 \\
\end{pmatrix}^n
\begin{pmatrix}
f_0 \\
f_1
\end{pmatrix}
= \begin{pmatrix}
f_n \\
f_{n+1}
\end{pmatrix}
\]

Gries and Levin/ Computing a Fibonacci number in log time. IPL 2 (October 1980), 68-69.

Constant-time algorithm!

```
Define $\phi = (1 + \sqrt{5}) / 2$  \quad $\phi' = (1 - \sqrt{5}) / 2$

The golden ratio again.

Prove by induction on $n$ that

\[ f_n = (\phi^n - \phi'^n) / \sqrt{5} \]

We went from $O(2^n)$ to $O(1)$