ICS141: Discrete Mathematics for Computer Science I

Dept. Information & Computer Sci., University of Hawaii

Jan Stelovsky
based on slides by Dr. Baek and Dr. Still
Originals by Dr. M. P. Frank and Dr. J.L. Gross
Provided by McGraw-Hill

Quiz

1. State the 1st Principle of Mathematical Induction
2. What is the difference between the 1st, 2nd and strong principles of Mathematical Induction. (Describe in plain English)
3. What is the big-O complexity of Euclid's Algorithm?

Lecture 21

Chapter 4. Induction and Recursion

4.3 Recursive Definitions and Structural Induction

Recursive Definitions

- In induction, we *prove* all members of an infinite set satisfy some predicate *P* by:
  - proving the truth of the predicate for larger members in terms of that of smaller members.
- In ***recursive definitions***, we similarly *define* a function, a predicate, a set, or a more complex structure over an infinite domain (universe of discourse) by:
  - defining the function, predicate value, set membership, or structure of larger elements in terms of those of smaller ones.

Recursion

- ***Recursion*** is the general term for the practice of defining an object in terms of *itself*
  - or of part of itself.
  - This may seem circular, but it isn't necessarily.
- An inductive proof establishes the truth of $P(k+1)$ *recursively* in terms of $P(k)$.
- There are also recursive *algorithms*, *definitions*, *functions*, *sequences*, *sets*, and other structures.

Recursively Defined Functions

University of Hawaii

- Simplest case: One way to define a function $f{:}\mathbf{N}\rightarrow S$ (for any set S) or series $a_n = f(n)$ is to:
  - Define $f(0)$
  - For $n > 0$, define $f(n)$ in terms of $f(0),\ldots,f(n-1)$
- **Example**: Define the series $a_n = 2^n$ where n is a nonnegative integer recursively:
  - a_n looks like $2^0, 2^1, 2^2, 2^3,\ldots$
  - Let $a_0 = 1$
  - For $n > 0$, let $a_n = 2\cdot a_{n-1}$

Another Example

University of Hawaii

- Suppose we define $f(n)$ for all $n \in \mathbf{N}$ recursively by:
  - Let $f(0) = 3$
  - For all $n > 0$, let $f(n) = 2 \cdot f(n-1) + 3$

- What are the values of the following?
  - $f(1) =$ $2 \cdot f(0) + 3 = 2 \cdot 3 + 3 = 9$
  - $f(2) =$ $2 \cdot f(1) + 3 = 2 \cdot 9 + 3 = 21$
  - $f(3) =$ $2 \cdot f(2) + 3 = 2 \cdot 21 + 3 = 45$
  - $f(4) =$ $2 \cdot f(3) + 3 = 2 \cdot 45 + 3 = 93$

Recursive Definition of Factorial

University of Hawaii

- Give an inductive (recursive) definition of the factorial function,

  $$F(n) = n! = \prod_{1 \le i \le n} i = 1 \cdot 2 \cdots n$$

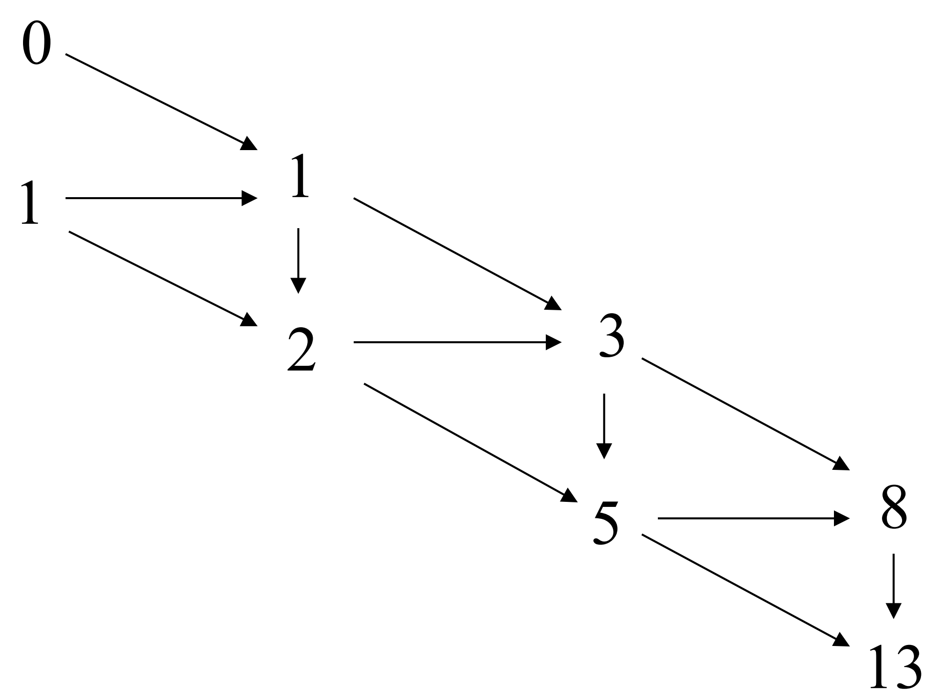
  - Basis step: $F(1) = 1$
  - Recursive step: $F(n) = n \cdot F(n-1)$ for $n > 1$
    - $F(2) = 2 \cdot F(1) = 2 \cdot 1 = 2$
    - $F(3) = 3 \cdot F(2) = 3 \cdot \{2 \cdot F(1)\} = 3 \cdot 2 \cdot 1 = 6$
    - $F(4) = 4 \cdot F(3) = 4 \cdot \{3 \cdot F(2)\} = 4 \cdot \{3 \cdot 2 \cdot F(1)\}$

      $= 4 \cdot 3 \cdot 2 \cdot 1 = 24$

The Fibonacci Numbers

- The ***Fibonacci numbers*** $f_{n\geq 0}$ is a famous series defined by:

$$f_0 = 0, \quad f_1 = 1, \quad f_{n\geq 2} = f_{n-1} + f_{n-2}$$



Inductive Proof about Fibonacci Numbers

- **Theorem:** $f_n < 2^n$. ← Implicitly for all $n \in \mathbf{N}$
- **Proof:** By induction
  - Basis step: $f_0 = 0 < 2^0 = 1$
    $f_1 = 1 < 2^1 = 2$
    Note: use of base cases of recursive definition
  - Inductive step: Use 2nd principle of induction (strong induction).

    Assume $\forall 0 \le i \le k,\ f_i < 2^i$. Then

    $$f_{k+1} = f_k + f_{k-1} \text{ is}$$
    $$< 2^k + 2^{k-1}$$
    $$< 2^k + 2^k = 2^{k+1}. \quad \blacksquare$$

A Lower Bound on Fibonacci Numbers

University of Hawaii

- **Theorem:** For all integers $n \geq 3$, $f_n > \alpha^{n-2}$, where $\alpha = (1 + 5^{1/2})/2 \approx 1.61803$.
- **Proof.** (Using strong induction.)
  - Let $P(n) = (f_n > \alpha^{n-2})$.
  - **Basis step**:

    For $n = 3$, note that $\alpha^{n-2} = \alpha < 2 = f_3$.

    For $n = 4$, $\alpha^{n-2} = \alpha^2$

    $$= (1 + 2 \cdot 5^{1/2} + 5)/4$$
    $$= (3 + 5^{1/2})/2$$
    $$\approx 2.61803 \qquad (= \alpha + 1)$$
    $$< 3 = f_4.$$

A Lower Bound on Fibonacci Numbers: Proof Continues…

- **Inductive step:** For $k \geq 4$, assume $P(j)$ for $3 \leq j \leq k$, prove $P(k+1)$.
  - $f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3}$ (by inductive hypothesis, $f_{k-1} > \alpha^{k-3}$ and $f_k > \alpha^{k-2}$).
  - Note that $\alpha^2 = \alpha + 1$.
    since $(3 + 5^{1/2})/2 = (1 + 5^{1/2})/2 + 1$
  - Thus, $\alpha^{k-1} = \alpha^2\alpha^{k-3} = (\alpha + 1)\alpha^{k-3}$
    $= \alpha\alpha^{k-3} + \alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}$.
  - So, $f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}$.
  - Thus $P(k+1)$. ■

University of Hawaii

Recursively Defined Sets

University of Hawaii

- An infinite set *S* may be defined recursively, by giving:
  - A small finite set of *base* elements of *S*.
  - A rule for constructing new elements of *S* from previously-established elements.
  - Implicitly, *S* has no other elements but these.

base element (basis step)

construction rule (recursive step)

- **Example:** Let $\underbrace{3 \in S}_{\text{base element (basis step)}}$, and let $\underbrace{x+y \in S \text{ if } x,y \in S}_{\text{construction rule (recursive step)}}$.
  What is *S*?

Example *cont.*

- Let $3 \in S$, and let $x+y \in S$ if $x,y \in S$. What is S?
  - $3 \in S$ (*basis step*)
  - 6 (= 3 + 3) is in S (*first application of recursive step*)
  - 9 (= 3 + 6) and 12 (= 6 + 6) are in S (*second application of the recursive step*)
  - 15 (= 3 + 12 or 6 + 9), 18 (= 6 + 12 or 9 + 9), 21 (= 9 + 12), 24 (= 12 + 12) are in S (*third application of the recursive step*)
  - … so on
  - Therefore, $S = \{3, 6, 9, 12, 15, 18, 21, 24, \ldots\}$
    = set of *all positive multiples of* 3