INDUCTION AND RECURSION

Climbing an Infinite Ladder

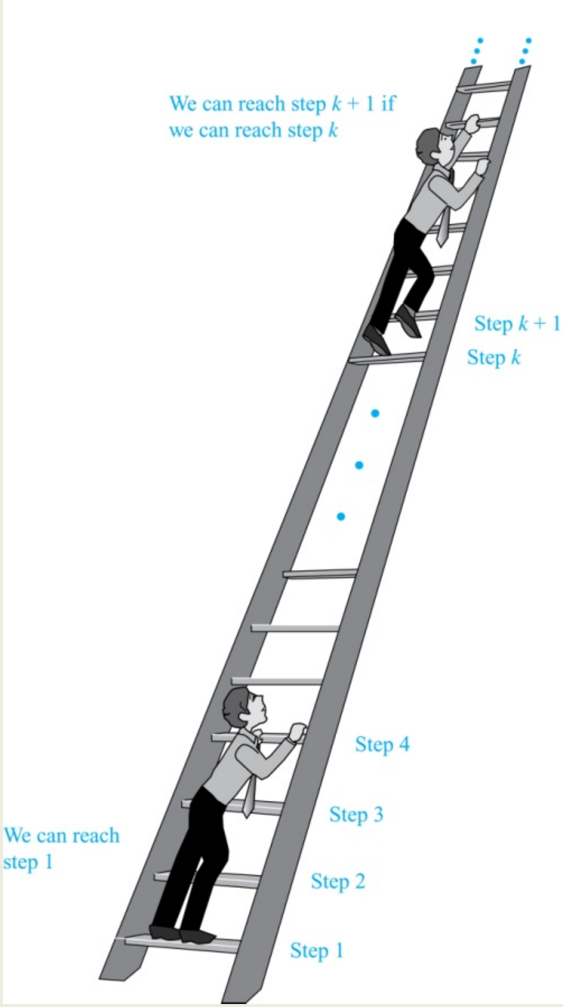
Suppose we have an infinite ladder:

1) We can reach the first rung of the ladder.
2) If we can reach a particular rung of the ladder, then we can reach the next rung.

From (1), we can reach the first rung.

By applying (2), we can reach the second rung. Applying (2) again, the third rung. And so on.

We can apply (2) any number of times to reach any particular rung, no matter how high up.



This example motivates proof by mathematical induction.

MATHEMATICAL INDUCTION (KR 5.1)

- Introduction
- Examples

Principle of Mathematical Induction

Prove that $P(n)$ is true for all positive integers n.

- ***Basis Step*:**
  Show that $P(1)$ is true.
- ***Inductive Step*:**
  Show that $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.

How?
Assuming the *inductive hypothesis* $P(k)$ holds for an arbitrary integer k, show that $P(k + 1)$ must be true.

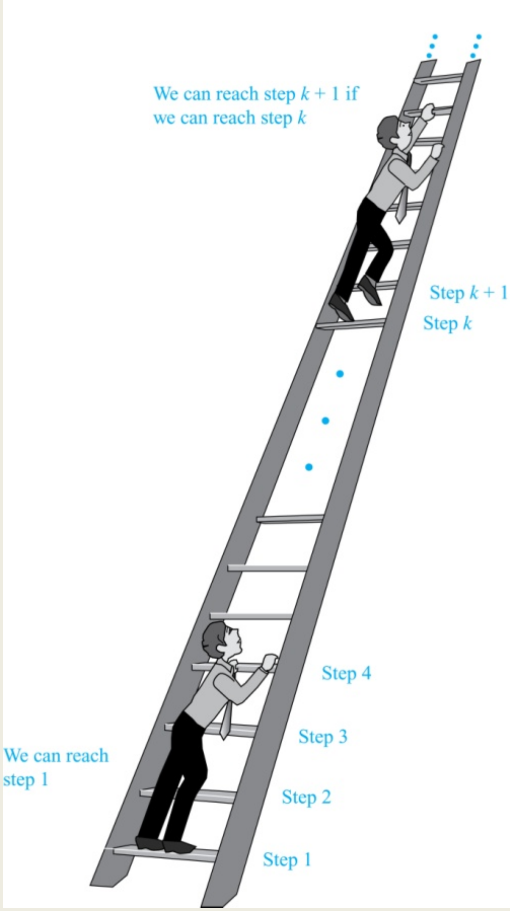
Principle of Mathematical Induction

Climbing an Infinite Ladder:

- Basis Step:
  By (1), we can reach rung 1.
- Inductive Step:
  Assume we can reach rung *k*.
  Then by (2), we can reach rung *k*+1.

Hence, $P(k) \rightarrow P(k+1)$ is true for all positive integers *k*.

We can reach every rung on the ladder.



Two Important Points About Using Mathematical Induction

1) In a proof by induction, we don't assume that $P(k)$ is true for all positive integers!

   We show: if we assume that $P(k)$ is true, then $P(k + 1)$ must also be true.

2) Proofs by mathematical induction do not always start at the integer 1.

   The basis step can begin at integer b, $b>1$.

Proving a Summation Formula by Induction

Claim: $\sum_{i=1}^{n} i = n(n+1)/2$

Proof:

BASIS STEP: *P*(1) is true since 1(1 + 1)/2 = 1.

INDUCTIVE STEP: Assume claim is true for *P*(*k*).

The inductive hypothesis is $\sum_{i=1}^{k} i = k(k+1)/2$

Under this assumption, show that P(k+1) is true:

$$1 + 2 + \ldots + k + (k+1) = \frac{k(k+1)}{2} + (k+1)$$

$$= \frac{k(k+1) + 2(k+1)}{2} \quad = \frac{(k+1)(k+2)}{2}$$

◀

Sum of the first n positive odd numbers

<u>Conjecture</u>: The sum of the first n positive odd integers is n^2.

We prove the conjecture correct using induction.

Inductive Hypothesis: $1 + 3 + 5 + \cdots + (2k - 1) = k^2$

BASIS STEP: $P(1)$ is true since $1^2 = 1$.

INDUCTIVE STEP: $P(k) \rightarrow P(k + 1)$ for every positive integer k.

Inductive Hypothesis: $1 + 3 + 5 + \cdots + (2k - 1) = k^2$

Under this assumption, show that P(k+1) is true:

$$
\begin{aligned}
1 + 3 + \cdots + (2k - 1) + (2k + 1) &= [1 + 3 + \cdots + (2k - 1)] + (2k + 1) \\
&= k^2 + (2k + 1) \quad (\textit{by the induct. hyp.}) \\
&= k^2 + 2k + 1 \\
&= (k + 1)^2
\end{aligned}
$$

Hence, we have shown that $P(k + 1)$ follows from $P(k)$. Therefore the sum of the first n positive odd integers is n^2.

Proving Inequalities

Example 1: Use induction to prove that $n < 2^n$ for all positive integers n.

Solution: Let $P(n)$ be the proposition that $n < 2^n$.

Proving Inequalities

Example 1: Use induction to prove that $n < 2^n$ for all positive integers n.

Solution: Let $P(n)$ be the proposition that $n < 2^n$.

BASIS STEP: $P(1)$ is true since $1 < 2^1 = 2$.

INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $k < 2^k$, for every positive integer k.

To shows that $P(k + 1)$ holds.

Since by the inductive hypothesis, $k < 2^k$, it follows that:

$$\boldsymbol{k+1} < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = \boldsymbol{2^{k+1}}$$

Therefore $n < 2^n$ holds for all positive integers n. ◀

Proving Inequalities

Example 2: Use induction to prove that $2^n < n!$, for every integer $n \geq 4$.

Solution: Let $P(n)$ be the proposition that $2^n < n!$

BASIS STEP: $P(4)$ is true since $2^4 = 16 < 4! = 24$.

INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $2^k < k!$ for every integer $k \geq 4$. To show that $P(k + 1)$ holds:

$$
\begin{aligned}
2^{k+1} &= 2 \cdot 2^k \\
&< 2 \cdot k! \qquad \textit{(by the inductive hypothesis)} \\
&< (k + 1)k! \\
&= (k + 1)!
\end{aligned}
$$

Therefore, $2^n < n!$ holds, for every integer $n \geq 4$. ◀

Note that the basis step is $P(4)$; $P(0)$, $P(1)$, $P(2)$, and $P(3)$ are false.

Example: Use induction to prove that $n^3 - n$ is divisible by 3, for every positive integer n.

Solution: Let $P(n)$ be the proposition that $n^3 - n$ is divisible by 3.

BASIS STEP: $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.

INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $\boldsymbol{k^3 - k}$ **is divisible** by 3, for every positive integer k.

Example: Use induction to prove that $n^3 - n$ is divisible by 3, for every positive integer n.

Solution: Let $P(n)$ be the proposition that $n^3 - n$ is divisible by 3.

BASIS STEP: $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.

INDUCTIVE STEP: Assume $P(k)$ holds, i.e., $\boldsymbol{k^3 - k}$ **is divisible** by 3, for every positive integer k.

Show that $P(k + 1)$ follows:

$$(k + 1)^3 - (k + 1) = (k^3 + 3k^2 + 3k + 1) - (k + 1)$$
$$= (k^3 - k) + 3(k^2 + k)$$

By inductive hypothesis, $(k^3 - k)$ is divisible by 3. The second term is divisible by 3 since it a multiple of 3. So, $(k + 1)^3 - (k + 1)$ is divisible by 3.

Therefore, $n^3 - n$ is divisible by 3.

Number of Subsets of a Finite Set

Example: Use induction to show that if S is a finite set with n elements, then S has 2^n subsets.

S = {1, 2, 3, 4} has $2^4 = 16$ subsets

Solution: Let $P(n)$ be the proposition that a set with n elements has 2^n subsets.

BASIS STEP: $P(0)$ is true, because the empty set has only itself as a subset and $2^0 = 1$.

INDUCTIVE STEP: Assume $P(k)$ is true for an arbitrary nonnegative integer k.

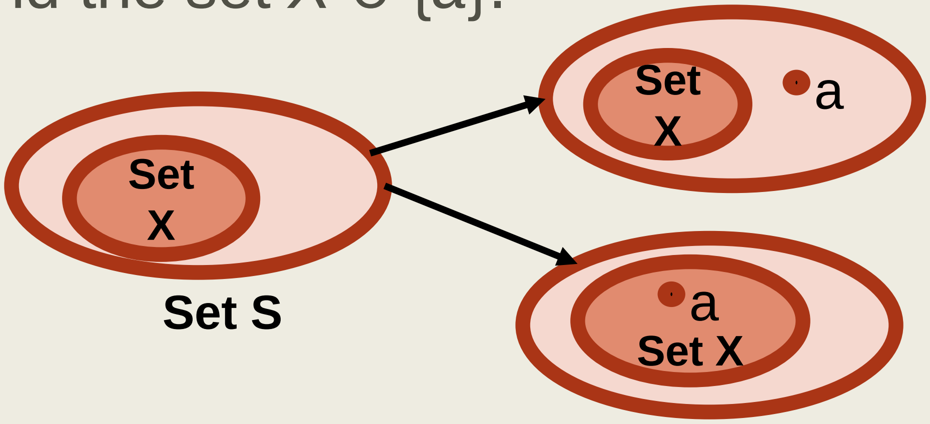
continued →

Inductive Hypothesis: For an arbitrary nonnegative integer k, every set with k elements has 2^k subsets.

Let T be a set with $k + 1$ elements.

Then, $T = S \cup \{a\}$, $a \in T$ and $S = T - \{a\}$. Hence $|S| = k$.

For each subset X of S, there are exactly two subsets of T: *set* X and the set $X \cup \{a\}$.

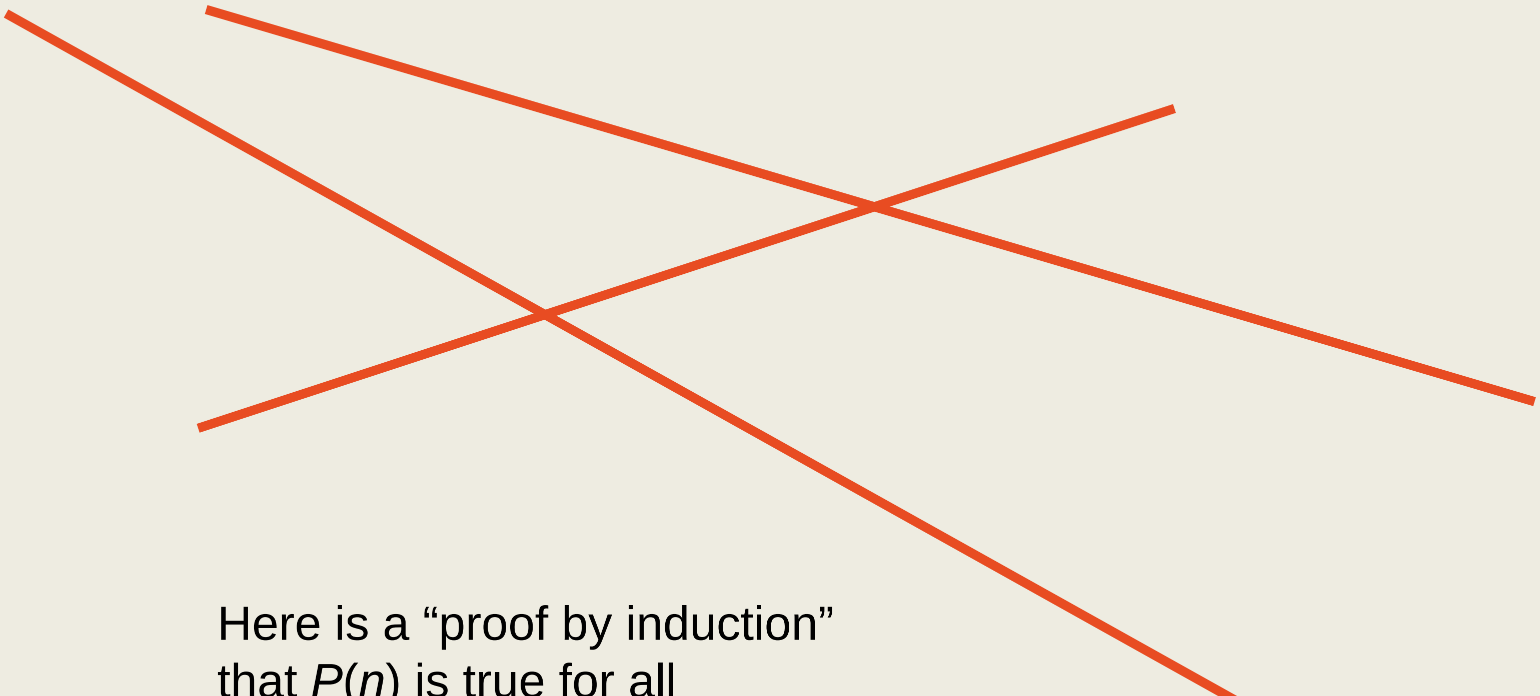


$T = S \cup \{a\}$

By the inductive hypothesis, set S has 2^k subsets. Since there are two subsets of T for each subset of S, the number of subsets of T is $2 \cdot 2^k = 2^{k+1}$.

An Incorrect "Proof" by Induction

Example: Let $P(n)$ be the statement that every set of n lines in the plane, no two of which are parallel, meet in a common point.



Here is a "proof by induction" that $P(n)$ is true for all positive integers $n \geq 2$.

continued →

An Incorrect “Proof” by Induction

Example: Let $P(n)$ be the statement that every set of n lines in the plane, no two of which are parallel, meet in a common point.

BASIS STEP: The statement $P(2)$ is true.
Any two lines in the plane that are not parallel meet in a common point.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true for the positive integer $k \geq 2$:
Every set of k lines in the plane, no two of which are parallel, meet in a common point.

We show that if $P(k)$ holds, then $P(k + 1)$ holds, i.e., if every set of k lines in the plane, no two of which are parallel, $k \geq 2$, meet in a common point, then every set of k + 1 lines in the plane, no two of which are parallel, meet in a common point.

continued →

Inductive Hypothesis: Every set of k lines in the plane, where $k \geq 2$, no two of which are parallel, meet in a common point.

Consider a set of $k + 1$ distinct lines in the plane, no two parallel.

- By the inductive hypothesis, the first k of these lines meet in a common point p_1.
- By the inductive hypothesis, the last k of these lines meet in a common point p_2.

If p_1 and p_2 are different points, all lines containing both of them must be the same line since two points determine a line. This contradicts the assumption that the lines are distinct.

Hence, $p_1 = p_2$ lies on all $k + 1$ distinct lines, and therefore $P(k + 1)$ holds. Assuming that $k \geq 2$, distinct lines meet in a common point, then every $k + 1$ lines meet in a common point.

There must be an error in this proof since the conclusion is absurd.

Where is the error?

Answer: $P(k) \rightarrow P(k + 1)$ only holds for $k \geq 3$.

It is not the case that $P(2)$ implies $P(3)$.

The first two lines must meet in a common point p_1

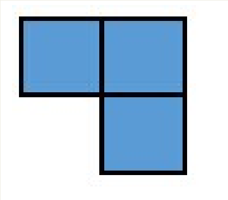
The second two lines must meet in a common point p_2.

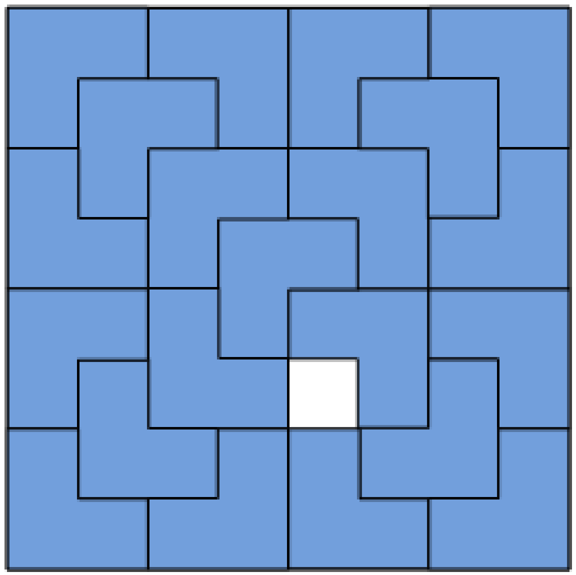
They **do not** have to be the same point since only the second line is common to both sets of lines.

Tiling Checkerboards

Example: Show that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes.

A right triomino is an L-shaped tile which covers three squares at a time.





A tiling of an 8 × 8 board

Show that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes.

Solution: Let $P(n)$ be the proposition that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes.

Use mathematical induction to prove that $P(n)$ is true for all positive integers n.

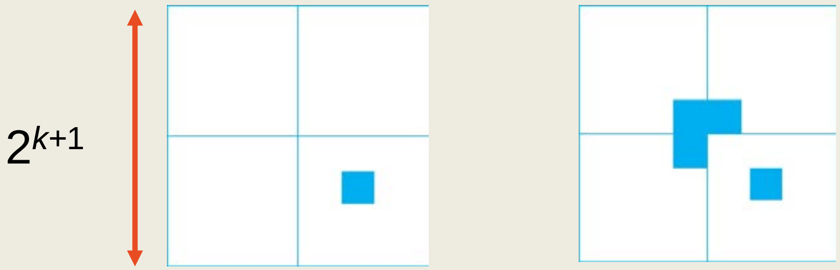
BASIS STEP: P(1) is true, because each of the four 2×2 checkerboards with one square removed can be tiled using one right triomino.

INDUCTIVE STEP: Assume that $P(k)$ is true for every $2^k \times 2^k$ board, for some positive integer k.

continued →

Consider a $2^{k+1} \times 2^{k+1}$ board with one square removed.

Split the board into four boards of size $2^k \times 2^k$



- By the inductive hypothesis, a $2^k \times 2^k$ board with the missing square can be tiled.
- Temporarily remove a corner square from each of the other three $2^k \times 2^k$ board as shown.
  - By the inductive hypothesis, each of these three boards can be tiled.
  - We place a right triominoe on the three removed corner squares.
- Hence, the entire $2^{k+1} \times 2^{k+1}$ checkerboard with one square removed can be tiled using right triominoes.

Mathematical Induction Proofs Template

Template for Proofs by Mathematical Induction

1. Express the statement that is to be proved in the form "for all $n \geq b$, $P(n)$" for a fixed integer b.
2. Write out the words "Basis Step." Then show that $P(b)$ is true, taking care that the correct value of b is used. This completes the first part of the proof.
3. Write out the words "Inductive Step."
4. State, and clearly identify, the inductive hypothesis, in the form "assume that $P(k)$ is true for an arbitrary fixed integer $k \geq b$."
5. State what needs to be proved under the assumption that the inductive hypothesis is true. That is, write out what $P(k + 1)$ says.
6. Prove the statement $P(k + 1)$ making use the assumption $P(k)$. Be sure that your proof is valid for all integers k with $k \geq b$, taking care that the proof works for small values of k, including $k = b$.
7. Clearly identify the conclusion of the inductive step, such as by saying "this completes the inductive step."
8. After completing the basis step and the inductive step, state the conclusion, namely that by mathematical induction, $P(n)$ is true for all integers n with $n \geq b$.

STRONG INDUCTION (KR 5.2)

- Definition
- Examples

Strong Induction

Strong Induction: To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, complete two steps:

BASIS STEP: Verify that the proposition $P(1)$ is true.

INDUCTIVE STEP: Show the conditional statement

$$[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k + 1)$$

holds for all positive integers k.

Strong Induction is sometimes called the *second principle of mathematical induction* or *complete induction*.

Which Form of Induction Should Be Used?

- We can always use strong induction instead of mathematical induction. But there is no reason to use it if it is simpler to use mathematical induction.
- In fact, the principles of mathematical induction, and strong induction are equivalent.

A Proof using Strong Induction (1)

Example: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

BASIS STEP: $P(12)$, $P(13)$, $P(14)$, and $P(15)$ hold.

$P(12)$ uses three 4-cent stamps.

$P(13)$ uses two 4-cent stamps and one 5-cent stamp.

$P(14)$ uses one 4-cent stamp and two 5-cent stamps.

$P(15)$ uses three 5-cent stamps.

INDUCTIVE STEP: The inductive hypothesis states that $P(j)$ holds for $12 \leq j \leq k$, where $k \geq 15$.

A Proof using Strong Induction (2)

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

INDUCTIVE STEP:

The inductive hypothesis states $P(j)$ holds for $12 \leq j \leq k$ with $k \geq 15$.

Assuming the inductive hypothesis, we show that $P(k + 1)$ holds.

A Proof using Strong Induction (2)

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

INDUCTIVE STEP:

The inductive hypothesis states $P(j)$ holds for $12 \leq j \leq k$ with $k \geq 15$.

Assuming the inductive hypothesis, we show that $P(k + 1)$ holds.

- Using the inductive hypothesis, $P(k - 3)$ holds since $k - 3 \geq 12$.
- To form postage of $k + 1$ cents, add a 4-cent stamp to the postage for $k - 3$ cents.

Hence, $P(n)$ holds for all $n \geq 12$.

Proof of Same Example using Mathematical Induction

Example: Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.

Solution: Let $P(n)$ be the proposition that postage of n cents can be formed using 4-cent and 5-cent stamps.

BASIS STEP: Postage of 12 cents can be formed using three 4-cent stamps.

INDUCTIVE STEP: The inductive hypothesis $P(k)$ for any positive integer k is that postage of k cents can be formed using 4-cent and 5-cent stamps.

To show $P(k + 1)$ where $k \geq 12$, consider two cases:

- If at least one 4-cent stamp has been used, then a 4-cent stamp can be replaced with a 5-cent stamp to yield a total of k + 1 cents.
- If no 4-cent stamp have been used, at least three 5-cent stamps were used. Three 5-cent stamps can be replaced by four 4-cent stamps to yield a total of k + 1 cents.

Hence, $P(n)$ holds for all $n \geq 12$.

RECURSIVE DEFINITIONS AND STRUCTURAL INDUCTION (KR 5.3)

- Introduction
- Examples

Recursively Defined Functions & Sequences

Define an function or sequence or set or object in terms of itself

Definition: A *recursive* or *inductive definition* of a function consists of two parts.

BASIS STEP: Specify the value of the function at zero.

RECURSIVE STEP: A rule for finding the value of f(n) from preceding function values.

A function $f(n)$ is the same as a sequence $a_0, a_1, \ldots, a_i \ldots$ where $f(i) = a_i$.

(See recurrence relations as seen in Section 2.4.)

Recursively Defined Functions

Suppose f is defined by:

$f(0) = 3$

$f(n + 1) = 2f(n) + 3$

Find $f(1)$, $f(2)$, $f(3)$

- $f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$
- $f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$
- $f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$

$f(n) = 2^n$

$f(0) = 1$

$f(n+1) = 2 \times f(n)$

$f(n) = n!$

$f(0) = 1$

$f(n + 1) = (n + 1) \times f(n)$

Set S (natural numbers)

$0 \in S$, $1 \in S$

If $a, b \in S$, then $a+b \in S$

Set T (elements are all powers of 2)

$1 \in T$

If $a \in T$, then $2a \in T$

Fibonacci Numbers

Fibonacci
(1170- 1250)

Example: The Fibonacci numbers are defined as:

$f_0 = 0$

$f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$

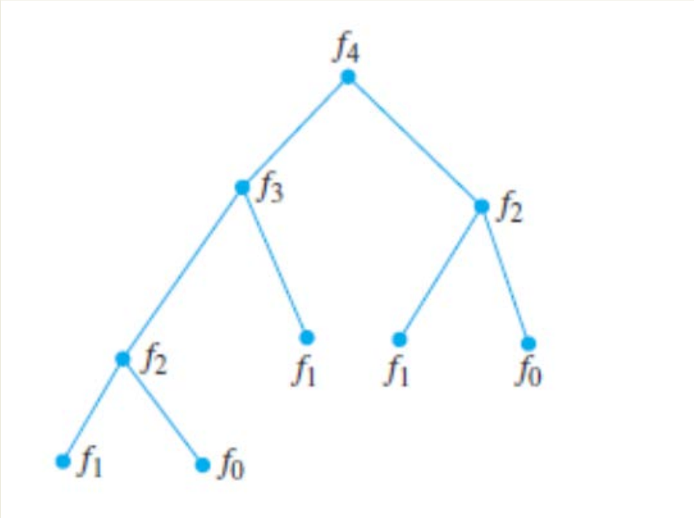
Find f_2, f_3 , f_4 , f_5

$f_2 = f_1 + f_0 = 1 + 0 = 1$

$f_3 = f_2 + f_1 = 1 + 1 = 2$

$f_4 = f_3 + f_2 = 2 + 1 = 3$

$f_5 = f_4 + f_3 = 3 + 2 = 5$



Fibonacci Numbers

Show that when $n \geq 3$, $f_n > \alpha^{n-2}$, where $\alpha = (1 + \sqrt{5})/2$.

Solution: Let $P(n)$ be the statement $f_n > \alpha^{n-2}$.

Use strong induction to show that $P(n)$ is true when $n \geq 3$.

BASIS STEP: $P(3)$ holds since $\alpha < 2 = f_3$

$P(4)$ holds since $\alpha^2 = (3 + \sqrt{5})/2 < 3 = f_4$

INDUCTIVE STEP:

Assume that $P(j)$ holds; i.e., $f_j > \alpha^{j-2}$ for all integers j with $3 \leq j \leq k$, where $k \geq 4$.

Show that $P(k + 1)$ holds, i.e., $f_{k+1} > \alpha^{k-1}$.

Show that $P(k+1)$ holds, i.e., $f_{k+1} > \alpha^{k-1}$

Since $\alpha^2 = \alpha + 1$ (because α is a solution of $x^2 - x - 1 = 0$),

$$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1) \cdot \alpha^{k-3} = \alpha \cdot \alpha^{k-3} + 1 \cdot \alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}$$

By the inductive hypothesis, because $k \geq 4$ we have

$$f_{k-1} > \alpha^{k-3}, \qquad f_k > \alpha^{k-2}.$$

Therefore, it follows that

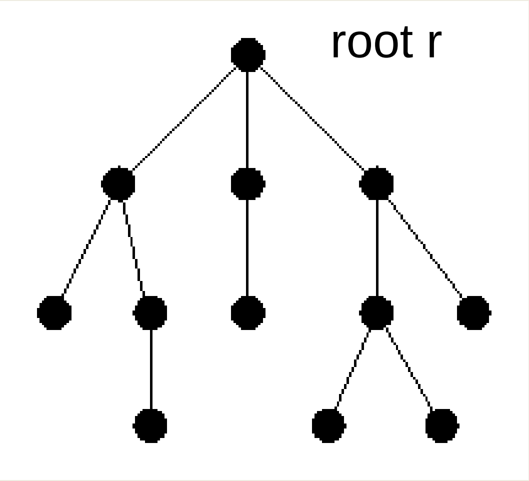
$$f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}.$$

Why does this equality hold?

Hence, $P(k+1)$ is true.

Rooted Trees

Definition: A rooted tree consists of a set of vertices containing a distinguished vertex called the *root*, and edges connecting these vertices.



A rooted tree can be defined recursively.

BASIS STEP: A single vertex r is a rooted tree.

RECURSIVE STEP: Suppose that T_1, T_2, …,T_n are disjoint rooted trees with roots r_1, r_2,…, r_n, respectively.

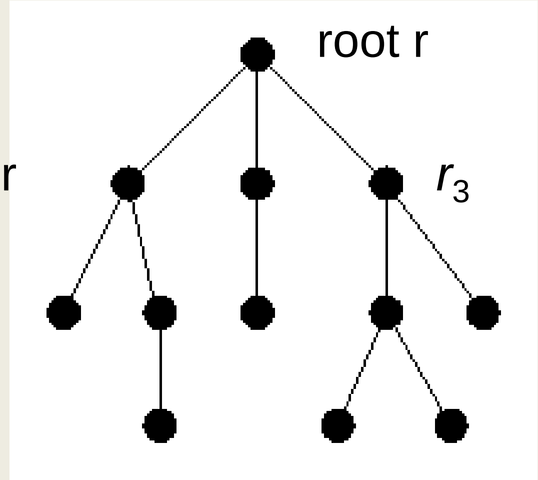
Rooted Trees

BASIS STEP: A single vertex r is a rooted tree.

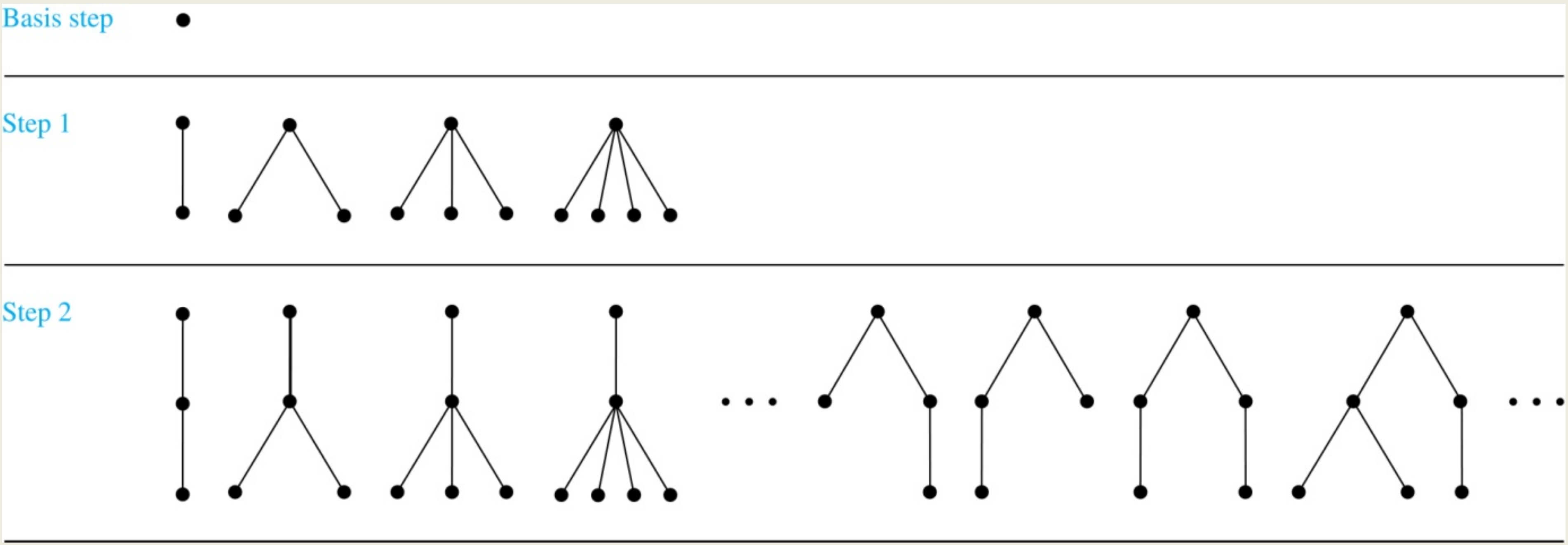
RECURSIVE STEP: Suppose that T_1, T_2, …,T_n are disjoint rooted trees with roots r_1, r_2,…, r_n, respectively.

The rooted tree is formed by starting with a root r, which is not in any of the rooted trees T_1, T_2, …,T_n, and adding an edge from r to each of the roots r_1, r_2,…, r_n.

children of root r



Building Up Rooted Trees



Trees are studied in Chapter 11.

Full Binary Trees

Definition: The set of *full binary trees* can be defined recursively by these steps.

BASIS STEP: A single vertex r is a full binary.

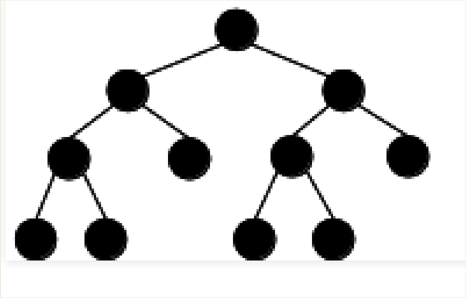
RECURSIVE STEP:

Assume T_1 and T_2 are disjoint full binary trees.

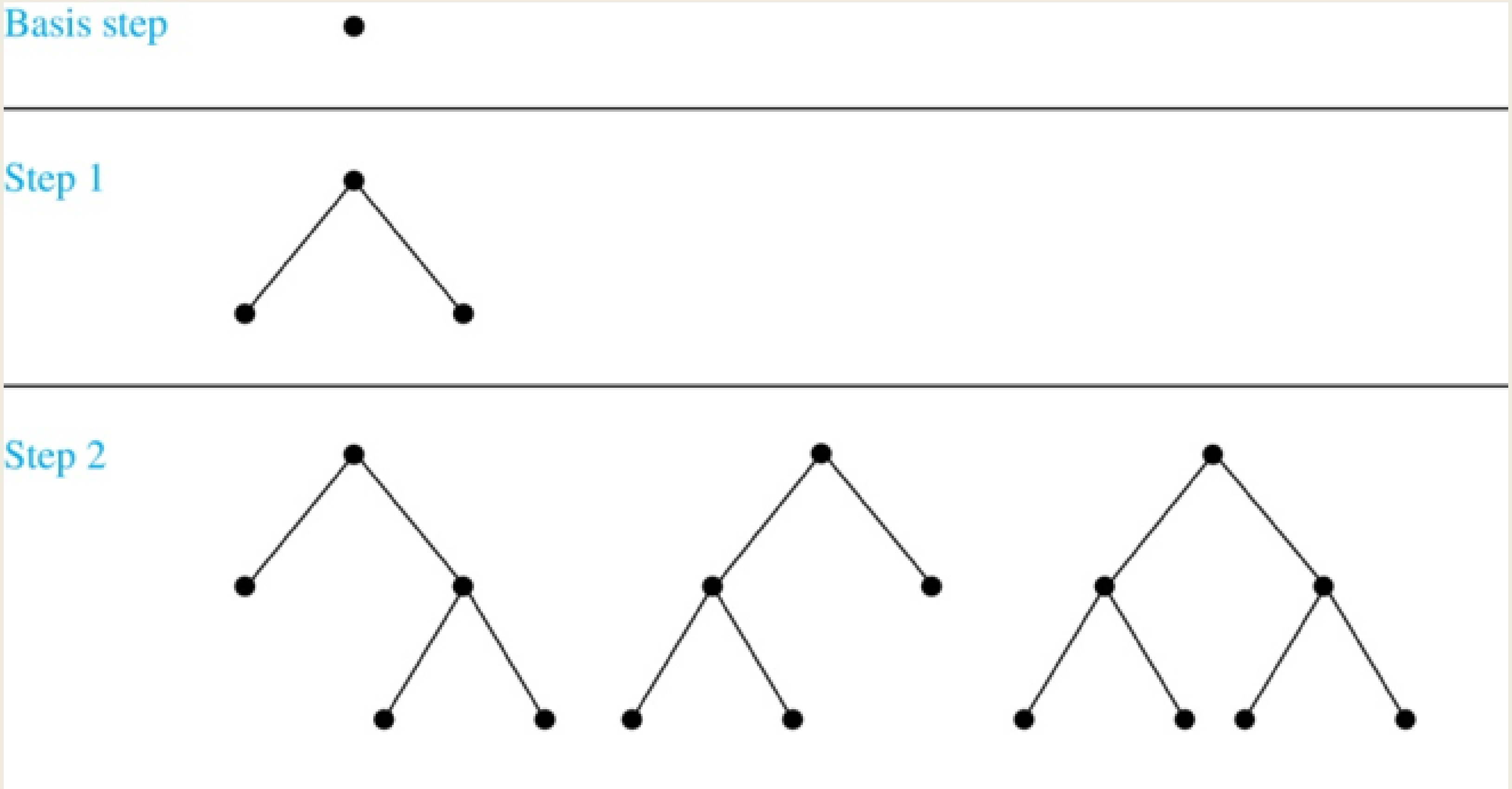
Then, there is a full binary tree T consisting of

- a root r
- an edge connecting r to the root of the left subtree T_1
- an edge connecting r to the root of the right subtree T_2.

Denote $T = T_1 \cdot T_2$



Building Up Full Binary Trees



Full binary trees and their properties play a central role in data structures and algorithms.

Structural Induction

Definition: To prove a property of the elements of a recursively defined set, we use *structural induction*.

BASIS STEP: Show that the result holds for all elements specified in the basis step of the recursive definition.

RECURSIVE STEP: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

The validity of structural induction can be shown to follow from the principle of mathematical induction.

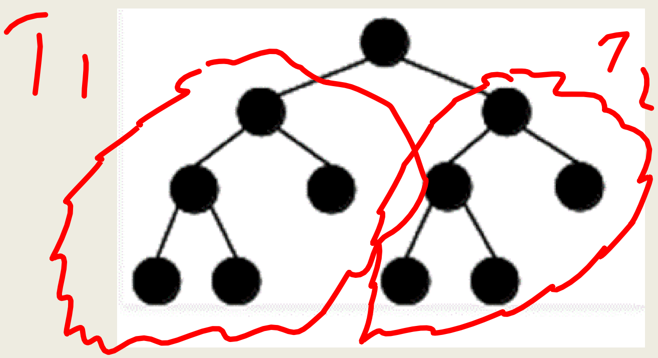
Full Binary Trees: Height

The ***height h(T)*** of a full binary tree *T* is defined recursively as follows:

BASIS STEP: The height of a full binary tree *T* consisting of only a root *r* is $h(T) = 0$.

$T = T_1 \cdot T_2$

RECURSIVE STEP:
If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has **height** $\boldsymbol{h(T) = 1 + \max\{h(T_1), h(T_2)\}}$.



Height(T) = 3

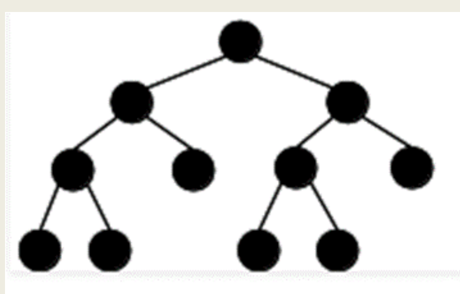
Full Binary Trees: Number of vertices

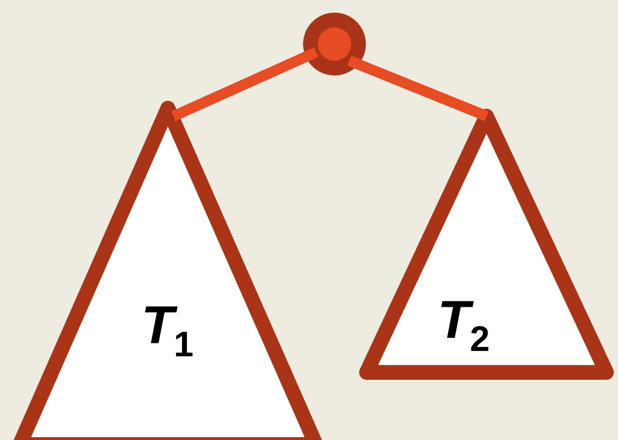
The **number of vertices** ***n*(*T*)** of a full binary tree T satisfies the following recursive formula:

BASIS STEP: The number of vertices of a full binary tree T consisting of only a root r is $n(T) = 1$.

RECURSIVE STEP:
If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has $\mathbf{n(T) = 1 + n(T_1) + n(T_2)}$ vertices.

n(T) = 5+5+1 = 11





Structural Induction and Binary Trees

Theorem: If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$.

Proof: Use structural induction. $1 + n(T_1) + n(T_2) \leq$

BASIS STEP: The result holds for a full binary tree consisting of only the root:

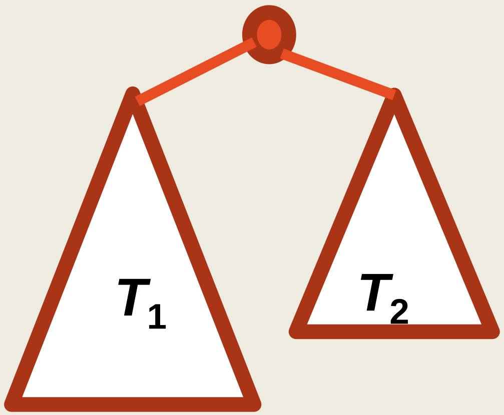
$n(T) = 1$ and $h(T) = 0$.

Hence, $n(T) = 1 \leq 2^{0+1} - 1 = 1$.

$2^{h(T_1)+1} - 1 + 2^{h(T_1)+1} - 1$

$+ 1$

RECURSIVE STEP: Let T_1 be the tree rooted at the left child of r and T_2 be the tree rooted at the right child.



RECURSIVE STEP:

As T_1 and T_2 are full binary trees,
we have $n(T_1) \le 2^{h(T1)+1} - 1$
and $n(T_2) \le 2^{h(T2)+1} - 1$.

$$
\begin{aligned}
\boldsymbol{n(T)} &= 1 + n(T_1) + n(T_2) && \textit{(by recursive formula of } n(T)\textit{)} \\
&\le \cancel{1} + (2^{h(T1)+1} - \cancel{1}) + (2^{h(T2)+1} - \cancel{1}) && \textit{(by induct. hyp.)} \\
&\le 2 \cdot \max\{2^{h(T1)+1}, 2^{h(T2)+1}\} - 1 \\
&= 2 \cdot 2^{\max\{h(T1), h(T2)\}+1} - 1 && (\max\{2^x, 2^y\} = 2^{\max\{x,y\}}) \\
&= 2 \cdot 2^{h(T)} - 1 && \textit{(by recursive definition of } h(T)\textit{)} \\
&= \boldsymbol{2^{h(T)+1} - 1}
\end{aligned}
$$

RECURSIVE ALGORITHMS (KR 5.4)

- Introduction
- Examples
- Recursion versus iteration

Recursive Algorithms

Definition: An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.

For the algorithm to terminate, the instance of the problem must eventually be reduced to some initial case for which the solution is known.

Recursive Factorial Algorithm

A recursive algorithm for computing *n*! using the recursive definition of the factorial function.

```
procedure factorial(n: nonnegative integer)
if  n=0 then return 1
else  return n·factorial(n−1)
{output is n!}
```

Why use recursion?

A loop from 1 to n would be faster (if you have to compute n! at all; 10! = 3,628,800).

Recursive Exponentiation Algorithm 1

A recursive algorithm for computing a^n, where a is a nonzero real number and n is a nonnegative integer.

Solution: Use the recursive definition of $a^n = aa^{n-1}$

```
procedure power1(a,n)

if  n = 0 then return 1
if  n = 1 then return a
else  return a*power1(a,n−1)
{output is aⁿ}
```

Recursive Exponentiation Algorithm 2

Another recursive definition of a^n

$$a^n = \begin{cases} a^{n/2}\, a^{n/2} & \text{when } n \text{ is even} \\ a\, a^{\lfloor n/2 \rfloor}\, a^{\lfloor n/2 \rfloor} & \text{when } n \text{ is odd} \end{cases}$$

```
procedure power2(a,n)
if  n = 0 then return 1
if  n = 1 then return a
if n is even
then return power2(a,n/2)*power2(a,n/2)
else return a*power2(a,⌊n/2⌋)*power2(a,⌊n/2⌋)
{output is aⁿ}
```

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & a \text{ is even} \\ a * a^{\lfloor n/2 \rfloor} * a^{\lfloor n/2 \rfloor} & a \text{ is odd.} \end{cases}$$

$$\log_5 m = \frac{\log_2 m}{\log_2 5}$$

Algorithm power1 versus power2

- Pseudocode of algorithm power1 is shorter
- Computing a^{15}
  - power1 makes 14 multiplications
  - power2 makes 2 + power2(7) multiplications:
    - power2(7)=4
    - total number of multiplications is 4+2=6
- Is algorithm power2 correct?
  - *power2(a,n/2)*power2(a,n/2)= (power2(a,n/2))*^{2}

Proving Recursive Algorithms Correct

Claim: Algorithm power2 computes a^n

Proof: Uses strong induction

- Basis step:
  - power2(a,0) = 1
  - power2(a,1) = a

- Inductive Step: $P(0), \ldots, P(k) \rightarrow P(k+1)$

Assume $P(i)$: $power2(a,i) = a^i$ for nonzero real a and all nonnegative integer $i \le k$

We show that $P(k+1)$: $power2(a,k+1) = a^{k+1}$

Case 1: k+1 is even

k+1 = 2j for nonnegative integer j

$$
\begin{aligned}
power2(a,k+1) &= power2(a,2j) = \\
&= power2(a,2j/2)^2 = \\
&= power2(a,j)^2 = (a^j)^2 = a^{2j} = \\
&= a^{k+1}
\end{aligned}
$$

Case 2: k+1 is odd

k+1 = 2j+1 for nonnegative integer j

$$
\begin{aligned}
power2(a,k+1) &= power2(a,2j+1) \\
&= a*power2(a,(2j+1-1)/2)^2 \\
&= a*power2(a,2j/2)^2 \\
&= a*power2(a,j)^2 \\
&= a*(a^j)^2 \\
&= a*a^{2j} = a^{2j+1} \\
&= a^{k+1}
\end{aligned}
$$

How many multiplications does power2 do?

Recurrence for number of multiplications for power2

$M(0) = M(1) = 0$

$M(n) = M(\lfloor n/2 \rfloor) + 1$ when n is even

$\quad\quad\quad M(\lfloor n/2 \rfloor) + 2$ when n is odd

$M(n) = O(\log(n))$

Recursive Linear Search (Algorithm 5 in KR 5.4)

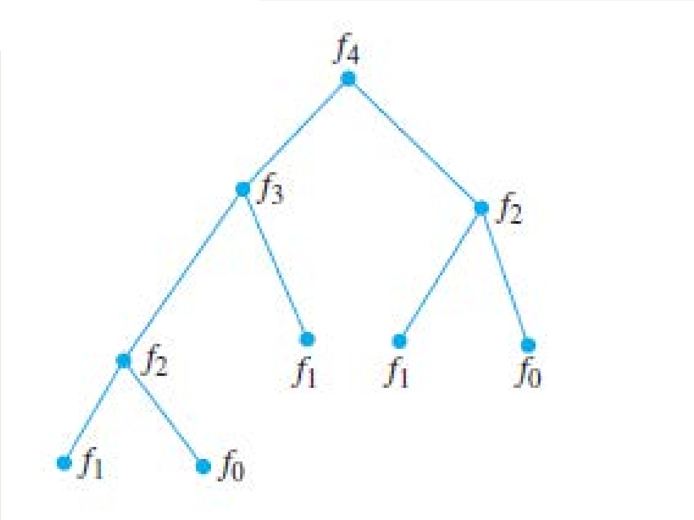
procedure *search*(i, j, x: i, j, x integers, $1 \leq i \leq j \leq n$)
if $a_i = x$ **then**
    return i
else if $i = j$ **then**
    return 0
else
    return *search*($i + 1, j, x$)
{output is the location of x in $a_1, a_2, \ldots, a_n$ if it appears; otherwise it is 0}

What is the benefit of recursion in this case?
None.

When to not use recursion?

Write a recursive algorithm to calculate the n^{th} Fibonacci number

procedure *fibonacci*(n: nonnegative integer)
if $n = 0$ **then return** 0
else if $n = 1$ **then return** 1
else return *fibonacci*($n - 1$) + *fibonacci*($n - 2$)
{output is *fibonacci*(n)}



Takes exponential time

Simple linear time solution exists

Recursive Binary Search

Array A contains n elements in non-decreasing order; search for x invoking binarySearch(1,n,x)

Procedure *binarySearch* (i, j, x; 1≤i ≤n, 1≤j ≤n)

m := $\lfloor (i+j)/2 \rfloor$

if x = A[m] **then** return(m)

else if (x < A[m] and i < m) **then**

    return**(***binarySearch*(i, m-1,x))

else if (x > A[m] and j > m) **then**

    return**(***binarySearch*(m+1,j,x))

else return(0)

Analysis of Recursive Binary Search

Example:

n=1023

1023 $\rightarrow$ 511 $\rightarrow$ 255 $\rightarrow$ 127 $\rightarrow$ 63 $\rightarrow$ 31 $\rightarrow$ 15 $\rightarrow$ 7 $\rightarrow$ 3 $\rightarrow$ 1

Number of comparisons made can be expressed as a recurrence relation:

Let floor(x) = $\lfloor x \rfloor$:

T(n) ≤ T(floor(n/2)) + 3, T(1) = 1, for worst case