Discrete Mathematics for Computer Science

Lecture 10: Mathematical Induction and Recursion

Dr. Ming Tang

Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Email: tangm3@sustech.edu.cn

SUSTech Southern University of Science and Technology

RSA Cryptosystem

RSA encryption: $C = M^e \bmod n$

RSA decryption: $M = C^d \bmod n$

- ...
- Divide this string into equally sized blocks of $2N$ digits
  - $2N$ is the largest even number such that the number 2525...25 with $2N$ digits does not exceed n.
- ....

Because $\gcd(p, q) = 1$, we have

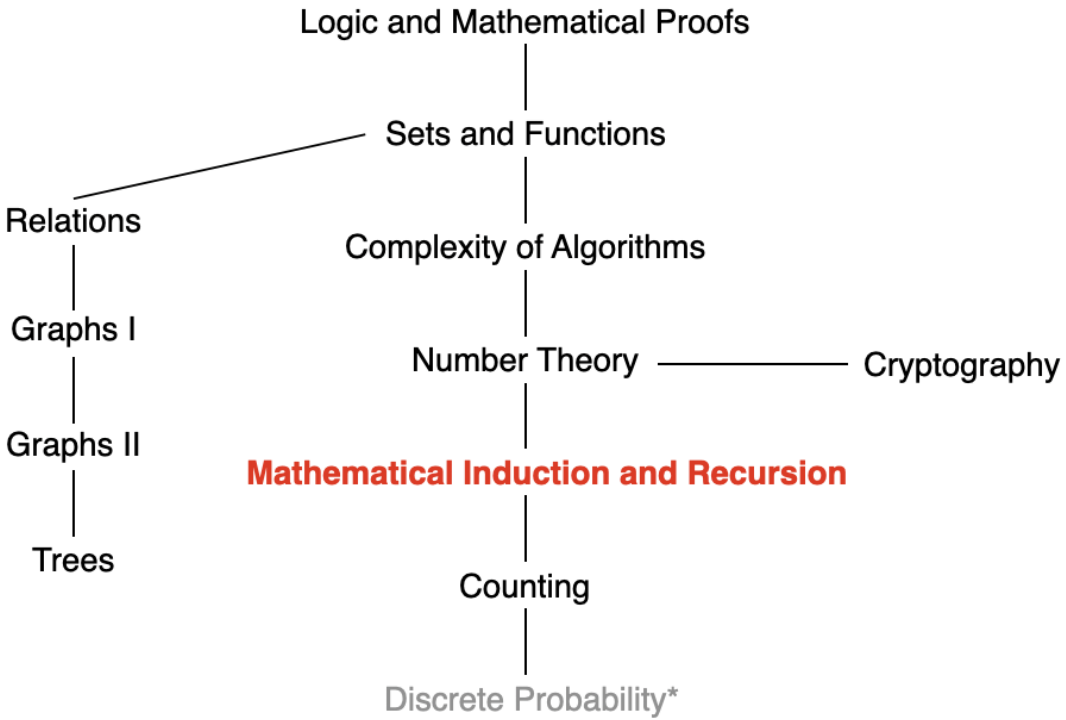
$$C^d \equiv M \pmod{pq}.$$

This basically implies that

$$M = C^d \bmod n$$

SUSTech Southern University of Science and Technology

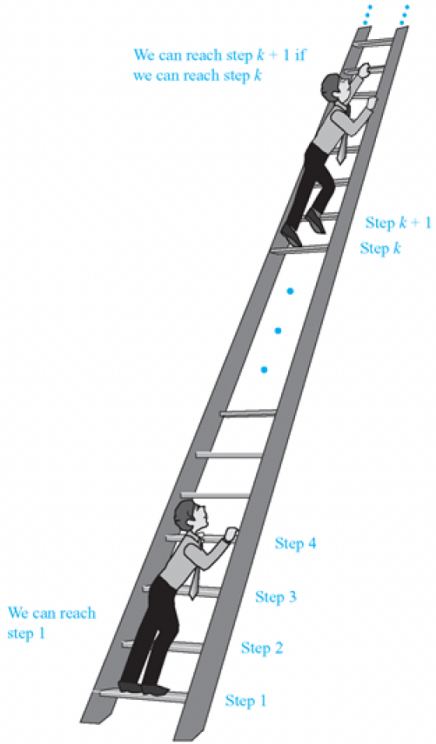
This Lecture

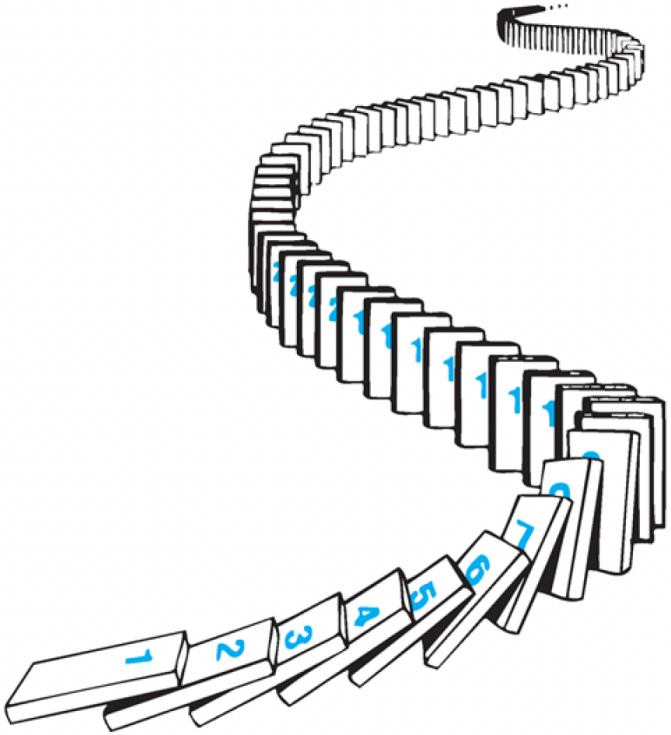


Mathematical induction, recursion, ...

SUSTech Southern University of Science and Technology

Mathematical Induction





SUSTech Southern University of Science and Technology

Mathematical Induction

We start by reviewing proof by smallest counterexample to try and understand what it is really doing.

This leads us to transform the indirect proof of proof by counterexample to direct proof. This direct proof technique will be **induction**.

We conclude by distinguishing between the weak principle of mathematical induction and the strong principle of mathematical induction.

- The strong principle can actually be derived from the weak principle.

SUSTech Southern University of Science and Technology

Mathematical Induction

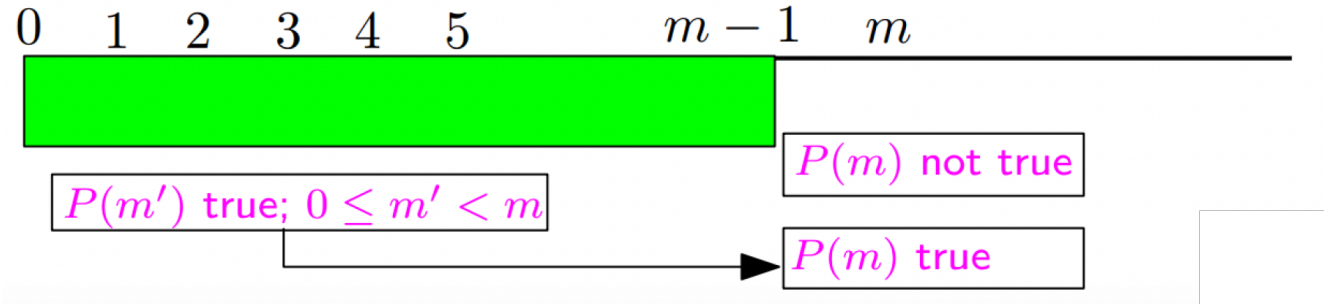
The statement $P(n)$ is true for all $n = 0, 1, 2, ...$

We prove this by

(i) Assume that a counterexample exists, i.e., There is some $n > 0$ for which $P(n)$ is false.

(ii) Let $m > 0$ be the smallest value for which $P(n)$ is false

(iii) Then, use the fact that $P(m')$ is true for all $0 \leq m' < m$ to show that $P(m)$ is true, contradicting the choice of m.



Contradiction!

Example 1

Use proof by smallest counterexample to show that, $\forall n \in N$,

$$0 + 1 + 2 + 3 + ... + n = \frac{(n+1)n}{2}.$$

We use (*) to denote this equation.

- Suppose that (*) is not always true.
- Then there must be a smallest $n \in N$ such that (*) does not hold for n.
- For any nonnegative integer $i < n$,

$$0 + 1 + 2 + 3 + ... + i = \frac{(i+1)i}{2}$$

Note that since the equation always holds for 0 (because $0 = (0+1) \times 0/2$), The smallest counterexample n is larger than 0.

Example 1

We now have

(i) smallest counterexample n is greater than 0, and

(ii) (*) holds for $n-1$

Substituting $n-1$ for i gives

$$0+1+2+3+...+n-1=\frac{(n-1)n}{2}$$

Adding n to both sides gives

$$0+1+2+3+...+n-1+n=\frac{(n-1)n}{2}+n=\frac{(n+1)n}{2}.$$

Thus, n is not a counterexample. Contradiction!

Therefore, (*) holds for all positive integers n.

SUSTech Southern University of Science and Technology

Example 1: Insights

The key step were

- $P(0)$ is true such that the smallest counterexample exists
- proving that

$$P(n-1) \to P(n)$$

Recall that $P(n)$ is the statement

$$0+1+2+3+\ldots+n=\frac{(n+1)n}{2}.$$

SUSTech Southern University of Science and Technology

Example 2

Use proof by smallest counterexample to show that, $\forall n \in \mathbb{N}$,

$$2^{n+1} \geq n^2 + 2.$$

Let $P(n)$ denote $2^{n+1} \geq n^2 + 2$.

We start by assuming that the statement $\forall n P(n)$ is false.

When a universal quantifier is false, there must be some n for which it is false.

SUSTech Southern University of Science and Technology

Example 2

Let n be the smallest nonnegative integer for which $2^{n+1} \not\geq n^2 + 2$.

This means that, for all $i \in N$ with $i < n$,

$$2^{i+1} \geq i^2 + 2.$$

Since $2^{0+1} \geq 0^2 + 2$, we know that $n > 0$. Thus, $n - 1$ is a nonnegative integer less than n.

Then setting $i = n - 1$ gives

$$2^{(n-1)+1} \geq (n-1)^2 + 2$$

or

$$2^n \geq n^2 - 2n + 1 + 2 = n^2 - 2n + 3. \qquad (*)$$

SUSTech Southern University of Science and Technology

Example 2

From $(*)$, we are now given $2^n \geq n^2 - 2n + 3$.

Multiply both sides by 2, giving

$$2^{n+1} = 2 \cdot 2^n \geq 2 \cdot (n^2 - 2n + 3) = 2n^2 - 4n + 6.$$

To get a contradiction, we want to convert the right side into $n^2 + 2$ plus an additional nonnegative term.

Thus, we write

$$\begin{aligned} 2^{n+1} &\geq 2n^2 - 4n + 6 \\ &= (n^2 + 2) + (n^2 - 4n + 4) \\ &= n^2 + 2 + (n-2)^2 \\ &\geq n^2 + 2. \end{aligned}$$

contradiction!

SUSTech Southern University of Science and Technology

Example 2

Let $P(n)$ denote $2^{n+1} \geq n^2 + 2$. We just showed that

(a) $P(0)$ is true

(b) If $n > 0$, then $P(n-1) \rightarrow P(n)$

What did we do?

- Suppose there is some n for which $P(n)$ is false (*)
- Let n be the smallest counterexample
- From (a) $n > 0$, so $P(n-1)$ is true
- From (b), using direct inference, $P(n)$ is true
- This leads to contradiction.
- Thus, $P(n)$ is true for all $n \in N$.

SUSTech Southern University of Science and Technology

Example 2

Let $P(n)$ denote $2^{n+1} \geq n^2 + 2$. We just showed that

(a) $P(0)$ is true

(b) If $n > 0$, then $P(n-1) \rightarrow P(n)$

Main idea: We used proof by smallest counterexample to derive that $P(n)$ is true for all $n \in N$.

This is an indirect proof. Is it possible to prove this fact directly?

Since $P(0)$ and $P(n-1) \rightarrow P(n)$, we see that

$$P(0) \text{ implies } P(1),\ P(1) \text{ implies } P(2), \ldots$$

SUSTech Southern University of Science and Technology

The Principle of Mathematical Induction

Well-Ordering Property: Every nonempty set of nonnegative integers has a least element.

The well-ordering property permits us to assume that every set of nonnegative integers has a smallest element, allowing us to use the smallest counterexample.

This is actually equivalent to the principle of mathematical induction.

Principle. (Weak Principle of Mathematical Induction)

(a) Basic Step: the statement $P(b)$ is true

(b) Inductive Step: the statement $P(n-1) \to P(n)$ is true for all $n > b$

Thus, $P(n)$ is true for all integers $n \geq b$.

SUSTech Southern University of Science and Technology

Proof by Induction: Example 1

For all $n \geq 0$, $2^{n+1} \geq n^2 + 2$

Let $P(n)$ denote $2^{n+1} \geq n^2 + 2$.

(i) Note that for $n = 0$, $2^{0+1} = 2 \geq 2 = 0^2 + 2$, which is $P(0)$

(ii) Suppose that $n > 0$ and that $2^n \geq (n-1)^2 + 2$ (*)

$$\begin{aligned} 2^{n+1} &\geq 2(n-1)^2 + 4 \\ &= (n^2 + 2) + (n^2 - 4n + 4) \\ &= n^2 + 2 + (n-2)^2 \\ &\geq n^2 + 2 \end{aligned}$$

Hence, we have just proven that for $n > 0$, $P(n-1) \rightarrow P(n)$.

By mathematical induction, $\forall n \geq 0, 2^{n+1} \geq n^2 + 2$.

SUSTech Southern University of Science and Technology

Proof by Induction: Example 2

For all $n \geq 2$, $2^{n+1} \geq n^2 + 3$

Let $P(n)$ denote $2^{n+1} \geq n^2 + 3$.

(i) Note that for $n = 2$, $2^{2+1} = 8 \geq 7 = 2^2 + 3$, which is $P(2)$

(ii) Suppose that $n > 2$ and that $2^n \geq (n-1)^2 + 3$ (*)
(Inductive Hypothesis)

$$\begin{aligned} 2^{n+1} &\geq 2(n-1)^2 + 6 \\ &= n^2 + 3 + n^2 - 4n + 4 + 1 \\ &= n^2 + 3 + (n-2)^2 + 1 \\ &> n^2 + 3 \end{aligned}$$

Hence, we have just proven that for $n > 2$, $P(n-1) \to P(n)$.

By mathematical induction, $\forall n > 2$, $2^{n+1} \geq n^2 + 3$. (Inductive Conclusion)

SUSTech Southern University of Science and Technology

Another Form of Induction

We may have another form of direct proof as follows.

- First suppose that we have proof of $P(0)$
- Next suppose that we have a proof that, $\forall n > 0$,

$$P(0) \wedge P(1) \wedge P(2) \wedge ... \wedge P(n-1) \rightarrow P(n)$$

- Then, $P(0)$ implies $P(1)$
- $P(0) \wedge P(1)$ implies $P(2)$
- $P(0) \wedge P(1) \wedge P(2)$ implies $P(3)$ . . .

Iterating gives us a proof of $P(n)$ for all n.

SUSTech Southern University of Science and Technology

Strong Induction

Principle (Strong Principle of Mathematical Induction):

(a) Basic Step: the statement $P(b)$ is true

(b) Inductive Step: for all $n > b$, the statement

$$P(b) \wedge P(b+1) \wedge \ldots \wedge P(n-1) \rightarrow P(n) \text{ is true.}$$

Then, $P(n)$ is true for all integers $n \geq b$.

SUSTech Southern University of Science and Technology

Mathematical Induction

In practice, we do not usually explicitly distinguish between the weak and strong forms.

In reality, they are equivalent to each other in that the weak form is a special case of the strong form, and the strong form can be derived from the weak form.

SUSTech Southern University of Science and Technology

Summary

A typical proof by mathematical induction, showing that a statement $P(n)$ is true for all integers $n \geq b$ consists of three steps:

- Base Step: We show that $P(b)$ is true.
- Inductive Step: We then, $\forall n > b$, show either of the following

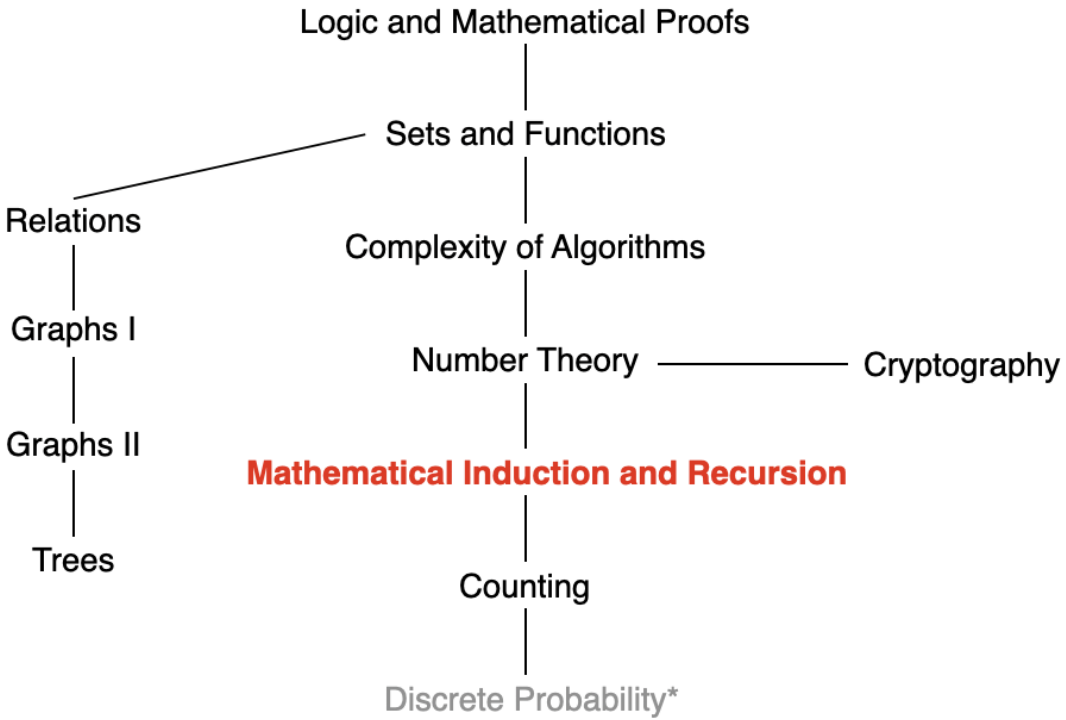
$$P(n-1) \rightarrow P(n)$$

$$P(b) \wedge P(b+1) \wedge ... \wedge P(n-1) \rightarrow P(n)$$

  We need to make the inductive hypothesis of either $P(n-1)$ or $P(b) \wedge P(b+1) \wedge ... \wedge P(n-1)$. We then derive $P(n)$.
- We conclude on the basis of the principle of mathematical induction that $P(n)$ is true for all $n \geq b$.

SUSTech Southern University of Science and Technology

This Lecture



Mathematical induction, recursion, ...

SUSTech Southern University of Science and Technology

Overview

- Towers of Hanoi
- Recurrence
- First-Order Linear Recurrence
- Growth Rates of Solutions to Recurrence

SUSTech Southern University of Science and Technology

Recursion

Recursive computer programs or algorithms often lead to inductive analysis.

A classical example of recursion is the Towers of Hanoi Problem.

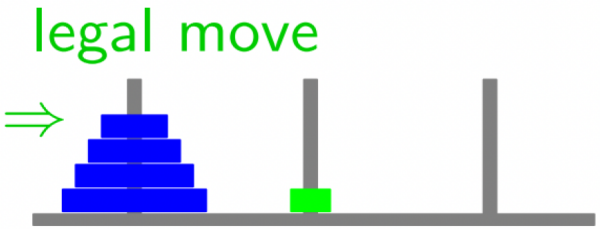
SUSTech Southern University of Science and Technology

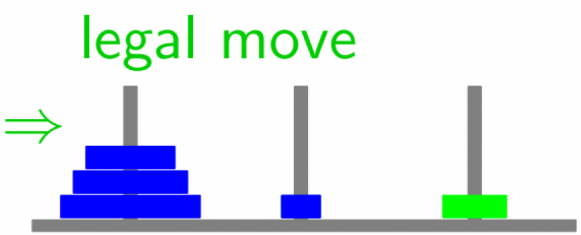
Towers of Hanoi

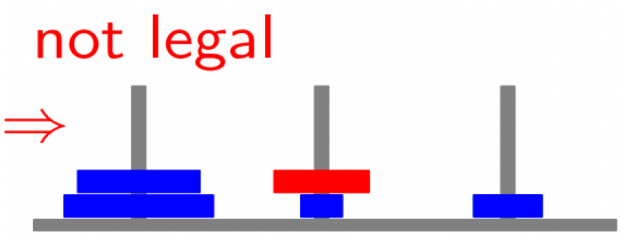
- 3 pegs, n disks of different sizes
- A legal move takes a disk from one peg and moves it onto another peg so that it is not on top of a smaller disk
- **Problem:** Find a (efficient) way to move all of the disks from one peg to another

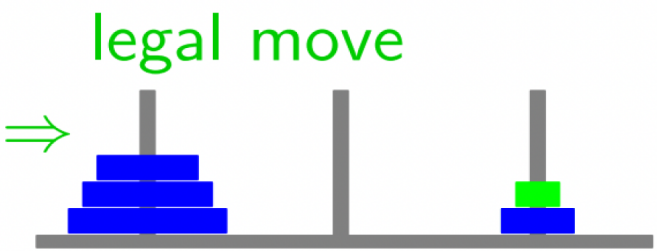
SUSTech Southern University of Science and Technology

Towers of Hanoi







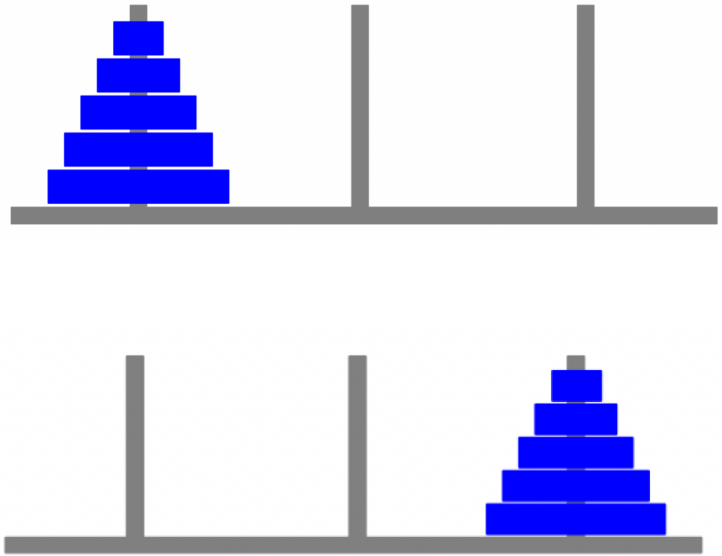


Towers of Hanoi

Problem: Start with n disks on leftmost peg

using only legal moves

move all disks to rightmost peg.



SUSTech Southern University of Science and Technology

Towers of Hanoi

Recursion Base:

If $n = 1$, moving one disk from i to j is easy. Just move it.

Towers of Hanoi

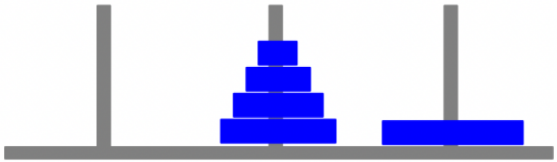
Given $i, j \in \{1, 2, 3\}$. Let $\overline{\{i, j\}} = \{1, 2, 3\} - \{i\} - \{j\}$. For example, $\overline{\{1, 2\}} = \{3\}$.

To move $n > 1$ disks from i to j

1)

move top $n - 1$ disks from i to $\overline{\{i, j\}}$

2)



move largest disk from i to j

3)

move top $n - 1$ disks from $\overline{\{i, j\}}$ to j

Towers of Hanoi

```
3  public class Hanoi
4  {
5
6      public void move(int n, char a, char b, char c)
7      {
8          if (n == 1)
9              System.out.println("plate " + n + " from " + a + " to " + c);
10         else
11         {
12             move(n-1,a,c,b);
13             System.out.println("plate " + n + " from " + a + " to " + c);
14             move(n-1,b,a,c);
15         }
16
17     }
18
```

SUSTech Southern University of Science and Technology

Towers of Hanoi

To prove correctness of solution, we are implicitly using induction:

$p(n)$ is statement that algorithm is correct for n

- $p(1)$ is statement that algorithm works for $n = 1$ disks, which is obviously true
- $p(n-1) \rightarrow p(n)$ is recursion statement that

  if our algorithm works for $n-1$ disks, then we can build a correct solution for n disks.

SUSTech Southern University of Science and Technology

Towers of Hanoi

Running Time: $M(n)$ is number of disk moves needed for n disks.

- $M(1) = 1$
- if $n > 1$, then $M(n) = 2M(n-1) + 1$

Iterating the recurrence gives $M(1) = 1$, $M(2) = 3$, $M(3) = 7$, $M(4) = 15$, $M(5) = 31$, . . .

We guess that $M(n) = 2^n - 1$.

We'll prove this by induction. Later, we'll also see how to solve without guessing.

SUSTech Southern University of Science and Technology

Towers of Hanoi

Formally, given

$$M(n) = \begin{cases} 1, & \text{if } n = 1 \\ 2M(n-1)+1, & \textit{otherwise} \end{cases}$$

We show that $M(n) = 2^n - 1$.

Proof. (by induction)

The base case $n = 1$ is true, since $2^1 - 1 = 1$.

For the inductive step, assume that $M(n-1) = 2^{n-1} - 1$ for $n > 1$.

Then $M(n) = 2M(n-1) + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 1$

SUSTech Southern University of Science and Technology

Towers of Hanoi

Note that we used induction twice.

- The first time was to derive correctness of algorithm and the recurrence

$$M(n) = \begin{cases} 1, & \text{if } n = 1 \\ 2M(n-1) + 1, & \textit{otherwise} \end{cases} \tag{1}$$

- The second time was to derive the closed form solution $M(n) = 2n - 1$ of the recurrence.

SUSTech Southern University of Science and Technology

Overview

- Towers of Hanoi
- Recurrence
- First-Order Linear Recurrence
- Growth Rates of Solutions to Recurrence

SUSTech Southern University of Science and Technology

Recurrence

A recurrence equation or recurrence for a function:

- defined on the set of integers $\geq b$
- tell us how to compute the n-th value from some or all the first $n-1$ values

To completely specify a function on the basis of a recurrence:

- Basis step (initial condition): Specify the value of the function at zero.
- Recursive step: Give a rule for finding its value at an integer from its values at smaller integers.

Example: $F(0) = 1$, $F(1) = 1$, $F(n) = F_{n-1} + F_{n-2}$ for $n \geq 2$.

SUSTech Southern University of Science and Technology

Recurrences: Example

Example 1: Let $S(n)$ be the number of subsets of a set of size n, i.e., $|\mathcal{P}(n)|$. What is the formula for $S(n)$?

The empty set (of size $n = 0$) has only one subset (itself), so $S(0) = 1$.

It is not difficult to see that $S(1) = 2$, $S(2) = 4$, $S(3) = 8$, ...

Based on counting, we have that $S(n) = 2^n$.

How to think it recursively?

SUSTech Southern University of Science and Technology

Recurrences: Example

Consider the subsets of $\{1, 2\}$:

$$\emptyset, \{1\}, \{2\}, \{1, 2\}$$

Consider the eight subsets of $\{1, 2, 3\}$:

$$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$$

It can be seen that

$$\begin{array}{cccc} \emptyset & \{1\} & \{2\} & \{1, 2\} \\ \{3\} & \{1, 3\} & \{2, 3\} & \{1, 2, 3\} \end{array}$$

This suggests that the recurrence for the number of subsets of an n-element set $\{1, 2, ..., n\}$ is $S(0) = 1$ and $S(n) = 2S(n-1)$ for $n \geq 1$.

SUSTech Southern University of Science and Technology

Iterating a Recurrence

Let $T(n) = rT(n-1) + a$, where r and a are constants.

Can we generalize this to find a closed-form solution?

$$\begin{aligned}
T(n) &= rT(n-1) + a \\
&= r(rT(n-2) + a) + a \\
&= r^2 T(n-2) + ra + a \\
&= r^2(rT(n-3) + a) + ra + a \\
&= r^3 T(n-3) + r^2 a + ra + a \\
&= r^3(rT(n-4) + a) + r^2 a + ra + a \\
&= r^4 T(n-4) + r^3 a + r^2 a + ra + a.
\end{aligned}$$

Guess $T(n) = r^n T(0) + a\sum_{i=0}^{n-1} r^i$.

SUSTech Southern University of Science and Technology

Iterating a Recurrence

The method we used to guess the solution is called iterating the recurrence, because we repeatedly (iteratively) use the recurrence.

Another approach is to iterate from the "bottom-up" instead of "top-down".

$$
\begin{aligned}
T(0) &= b \\
T(1) &= rT(0) + a = rb + a \\
T(2) &= rT(1) + a = r(rb + a) + a = r^2 b + ra + a \\
T(3) &= rT(2) + a = r^3 b + r^2 a + ra + a
\end{aligned}
$$

This would lead to the same guess:

$$T(n) = r^n b + a \sum_{i=0}^{n-1} r^i.$$

SUSTech Southern University of Science and Technology

Formula of Recurrences

Theorem: If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a\frac{1-r^n}{1-r}$$

for all nonnegative integers n.

How to prove?

Mathematical Induction:

- Basis step: We verify that $T(0)$ holds .
- Inductive step: We show that the conditional statement "if $T(n-1)$ holds, then $T(n)$ holds" for all $n \geq 1$.

SUSTech Southern University of Science and Technology

Formula of Recurrences

Theorem: If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a\frac{1-r^n}{1-r}$$

for all nonnegative integers n.

- Basis step: We verify that $T(0)$ holds:

$$T(0) = r^0 b + a\frac{1-r^0}{1-r}$$

- Inductive step: We show that the conditional statement "if $T(n-1)$ holds, then $T(n)$ holds" for all $n \geq 1$:

  Now assume that $n > 0$ and

$$T(n-1) = r^{n-1} b + a\frac{1-r^{n-1}}{1-r}.$$

SUSTech Southern University of Science and Technology

Formula of Recurrences

- Basis step: We verify that $T(0)$ holds:
- Inductive step: We show that the conditional statement "if $T(n-1)$ holds, then $T(n)$ holds" for all $n \geq 1$:
  Now assume that $n > 0$ and

$$T(n-1) = r^{n-1}b + a\frac{1-r^{n-1}}{1-r}.$$

Thus,

$$\begin{aligned} T(n) &= rT(n-1) + a \\ &= r\left(r^{n-1}b + a\frac{1-r^{n-1}}{1-r}\right) + a \\ &= r^n b + \frac{ar - ar^n}{1-r} + a \\ &= r^n b + \frac{ar - ar^n + a - ar}{1-r} \\ &= r^n b + a\frac{1-r^n}{1-r}. \end{aligned}$$

USTech Southern University of Science and Technology

Overview

- Towers of Hanoi
- Recurrences
- First-Order Linear Recurrences
- Growth Rates of Solutions to Recurrences

SUSTech Southern University of Science and Technology

First-Order Linear Recurrences

A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a first-order linear recurrence.

- **First Order**: because it only depends upon going back one step, i.e., $T(n-1)$
- If it depends upon $T(n-2)$, then it would be a second-order recurrence, e.g., $T(n) = T(n-1) + 2T(n-2)$.
- **Linear**: because $T(n-1)$ only appears to the first power.
- Something like $T(n) = (T(n-1))^2 + 3$ would be a non-linear first-order recurrence relation.

SUSTech Southern University of Science and Technology

First-Order Linear Recurrences

$$T(n) = f(n)T(n-1) + g(n)$$

When $f(n)$ is a constant, say r, the general solution is almost as easy as we derived before. Iterating the recurrence gives

$$\begin{aligned} T(n) &= rT(n-1) + g(n) \\ &= r(rT(n-2) + g(n-1)) + g(n) \\ &= r^2 T(n-2) + rg(n-1) + g(n) \\ &= r^3 T(n-3) + r^2 g(n-2) + rg(n-1) + g(n) \\ &\vdots \\ &= r^n T(0) + \sum_{i=0}^{n-1} r^i g(n-i) \end{aligned}$$

SUSTech Southern University of Science and Technology

First-Order Linear Recurrences

Theorem: For any positive constants a and r, and any function g defined on nonnegative integers, the solution to the first-order linear recurrence

$$T(n) = \begin{cases} rT(n-1) + g(n), & \text{if } n > 0 \\ a, & \text{if } n = 0 \end{cases}$$

is

$$T(n) = r^n a + \sum_{i=1}^{n} r^{n-i} g(i).$$

Proof by induction

SUSTech Southern University of Science and Technology

Example 1

Solve $T(n) = 4T(n-1) + 2^n$ with $T(0) = 6$.

$$\begin{aligned} T(n) &= 6 \cdot 4^n + \sum_{i=1}^{n} 4^{n-i} \cdot 2^i \\ &= 6 \cdot 4^n + 4^n \sum_{i=1}^{n} 4^{-i} \cdot 2^i \\ &= 6 \cdot 4^n + 4^n \sum_{i=1}^{n} (\frac{1}{2})^i \\ &= 6 \cdot 4^n + (1 - \frac{1}{2^n}) \cdot 4^n \\ &= 7 \cdot 4^n - 2^n. \end{aligned}$$

SUSTech Southern University of Science and Technology

Example 2

Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$.

$$\begin{aligned} T(n) &= 10 \cdot 3^n + \sum_{i=1}^{n} 3^{n-i} \cdot i \\ &= 10 \cdot 3^n + 3^n \sum_{i=1}^{n} i \cdot 3^{-i} \end{aligned}$$

Theorem. For any real number $x \neq 1$,

$$\sum_{i=1}^{n} i x^i = \frac{n x^{n+2} - (n+1)x^{n+1} + x}{(1-x)^2}.$$

SUSTech Southern University of Science and Technology

Example 2

Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$.

$$\begin{aligned} T(n) &= 10 \cdot 3^n + \sum_{i=1}^{n} 3^{n-i} \cdot i \\ &= 10 \cdot 3^n + 3^n \sum_{i=1}^{n} i \cdot 3^{-i} \\ &= 10 \cdot 3^n + 3^n \left(-\frac{3}{2}(n+1)3^{-(n+1)} - \frac{3}{4}3^{-(n+1)} + \frac{3}{4} \right) \\ &= \frac{43}{4}3^n - \frac{n+1}{2} - \frac{1}{4}. \end{aligned}$$

SUSTech Southern University of Science and Technology

Overview

- Towers of Hanoi
- Recurrences
- First-Order Linear Recurrences
- Growth Rates of Solutions to Recurrences

SUSTech Southern University of Science and Technology

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences
- Three different behaviors

SUSTech Southern University of Science and Technology

Divide and conquer algorithms

We just analyzed recurrences of the form

$$T(n) = \begin{cases} b, & \text{if } n = 0 \\ rT(n-1) + a, & \text{if } n > 0 \end{cases}$$

These corresponded to the analysis of recursive algorithms in which a problem of size n is solved by recursively solving a problem of size $n-1$.

We will now look at recurrences of the form

$$T(n) = \begin{cases} \text{something given}, & \text{if } n \leq n_0 \\ rT(n/m) + a, & \text{if } n > n_0 \end{cases}$$

SUSTech Southern University of Science and Technology

Example: Binary Search

Someone has chosen a number x between 1 and n. We need to discover x.

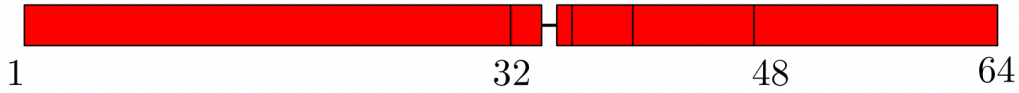
We are only allowed to ask two types of questions:

- Is x equal to k?
- Is x greater than k?

Our strategy will be to always ask greater than questions, at each step halving our search range, until the range only contains one number, when we ask a final equal to question.

SUSTech Southern University of Science and Technology

Example: Binary Search



Is $x > 32$?	Answer: Yes
Is $x > 48$?	Answer: No
Is $x > 40$?	Answer: No
Is $x > 36$?	Answer: No
Is $x > 34$?	Answer: Yes
Is $x > 35$?	Answer: No
Is $x = 35$?	Answer: BINGO!

SUSTech Southern University of Science and Technology

Example: Binary Search

Why this is a good approach?

Method: Each guess reduces the problem to one in which the range is only half as big.

This divides the original problem into one that is only half as big; we can now (recursively) conquer this smaller problem.

Note: When n is a power of 2, the number of questions in a binary search on $[1, n]$, satisfies

$$T(n) = \begin{cases} 1, & \text{if } n = 1 \\ T(n/2) + 1, & \text{if } n \geq 2 \end{cases}$$

This can also be proven inductively.

SUSTech Southern University of Science and Technology

Binary Search Example

$T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$

$$T(n) = \begin{cases} 1, & \text{if } n = 1 \\ T(n/2) + 1, & \text{if } n \geq 2 \end{cases}$$

The number of questions needed for binary search on n items is:

- first step
- time to perform binary search on the remaining $n = 2$ items

To compute the number of questions (suppose n is the power of 2)

- Basis step: $T(1) = 1$ to ask: "Is the number k?"
- Inductive step: $T(n) = T(n/2) + 1$

SUSTech Southern University of Science and Technology

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences
- Three different behaviors

SUSTech Southern University of Science and Technology