Discrete Mathematics for Computer Science

Lecture 11: Counting

Dr. Ming Tang

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

Last Lecture: Mathematical Induction

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

- Proof by contradiction: find the smallest counterexample
- **Well-Ordering Property:** Every nonempty set of nonnegative integers has a least element.
- **Weak Principle of Mathematical Induction**
  (a) Basic Step: the statement $P(b)$ is true
  (b) Inductive Step: the statement $P(n-1) \rightarrow P(n)$ is true for all $n > b$
- **Strong Principle of Mathematical Induction**
  (a) Basic Step: the statement $P(b)$ is true
  (b) Inductive Step: for all $n > b$, the statement

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

Last Lecture: Recursion

Towers of Hanoi:

- Move $n = 1$ disk; given how to move $n - 1$ disks, move n disks
- Prove the correctness of algorithm and running time
- Prove the closed-form solution of the running time

To obtain **recurrence equation:**

- 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: The number of subsets of a set of size n.

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

Last Lecture: Recursion

To obtain the closed-form solution:

- **Iterating a recurrence**:

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

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

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

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

  for all nonnegative integers n.

- **Proof by mathematical induction.**

Last Lecture: Recursion

First-order linear recurrence: $T(n) = f(n)T(n-1) + g(n)$

- **First Order**: $T(n-1)$
- **Linear**: because $T(n-1)$ only appears to the first power

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

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

Example 1

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

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

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

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 $x \neq 1$,

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

Example 2

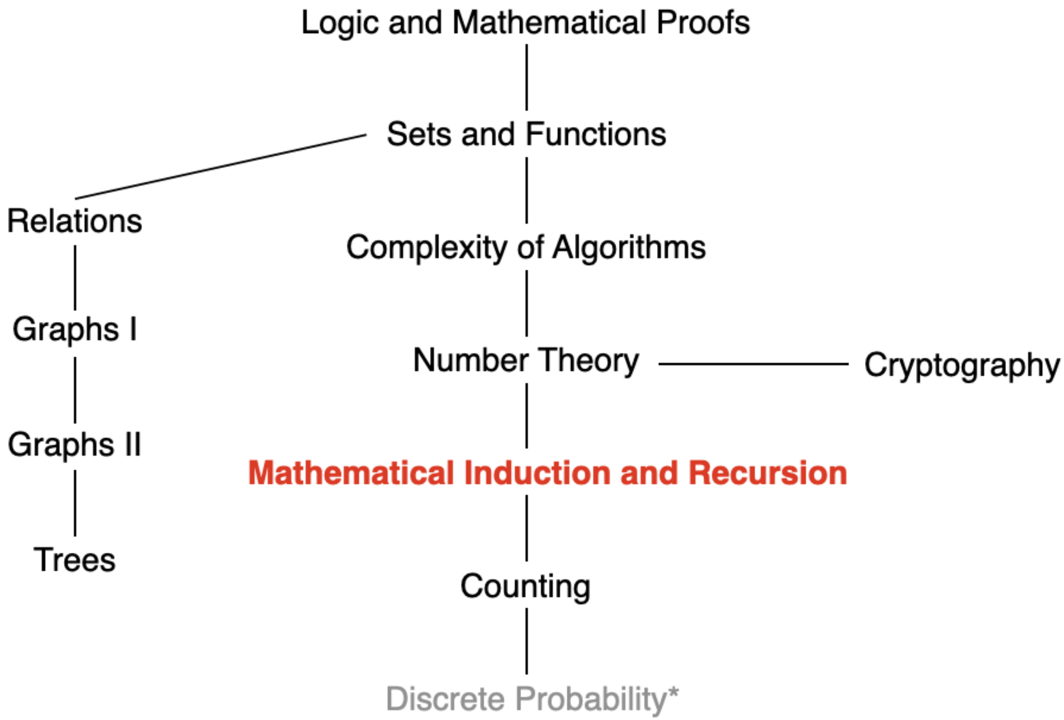
Theorem. For any real number $x \neq 1$,

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

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

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

This Lecture



Mathematical induction, recursion, ...

Recurrences of Another Form

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

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

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences (to obtain the closed-form)
- Three different behaviors

Divide and conquer algorithms

Such algorithms recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly

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.

Example: Binary Search

Why this is a good approach?

Each guess reduces the problem 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

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences (to obtain the closed-form)
- Three different behaviors

Iterating Recurrences: Example 1

Consider

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

This corresponds to solving a problem of size n:

- using $T(1)$ work for "bottom" case of $n = 1$
- solving 2 subproblems of size $n/2$ and doing n units of additional work

Iterating Recurrences: Example 1

Algebraically iterating the recurrence
(assume that n is a power of 2):

$$T(n) = 2T\left(\frac{n}{2}\right) + n \qquad = 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 4T\left(\frac{n}{4}\right) + 2n \qquad = 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n$$

$$\vdots \qquad \vdots$$

$$= 2^i T\left(\frac{n}{2^i}\right) + in$$

End when $i = \log_2 n$

$$\vdots \qquad \vdots$$

$$= 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + (\log_2 n)n$$

$$nT(1) + n\log_2 n$$

Iterating Recurrences: Example 1

We just iterated the recurrence to derive that the solution to

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

is $nT(1) + n \log_2 n$.

Note: Technically, we still need to use induction to prove that our solution is correct. Here, we do not explicitly perform this step, since it is obvious how the induction would work.

Iterating Recurrences: Example 2

Consider

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

$$\begin{aligned} T(n) &= T\left(\tfrac{n}{2}\right) + 1 && = \left(T\left(\tfrac{n}{2^2}\right) + 1\right) + 1 \\ &= T\left(\tfrac{n}{2^2}\right) + 2 && = \left(T\left(\tfrac{n}{2^3}\right) + 1\right) + 2 \\ &= T\left(\tfrac{n}{2^3}\right) + 3 \\ &\ \ \vdots \qquad \vdots \\ &= T\left(\tfrac{n}{2^i}\right) + i \\ &\ \ \vdots \qquad \vdots \\ &= T\left(\tfrac{n}{2^{\log_2 n}}\right) + \log_2 n \ = 1 + \log_2 n \end{aligned}$$

Iterating Recurrences: Example 3

Consider

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

$$\begin{aligned}
T(n) &= T\left(\frac{n}{2}\right) + n \\
&= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n \\
&= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n \\
&\quad \vdots \qquad \vdots \\
&= T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n \\
&\quad \vdots \qquad \vdots \\
&= T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{n}{2^{\log_2 n - 1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n \\
&= 1 + 2 + 2^2 + \cdots + \frac{n}{2^2} + \frac{n}{2} + n
\end{aligned}$$

$\Theta(n)$

Iterating Recurrences: Example 4

Consider

$$T(n) = \begin{cases} 1, & \text{if } n < 3, \\ 3T(n/3) + n, & \text{if } n \geq 3. \end{cases}$$

$$\begin{aligned}
T(n) &= 3T\left(\tfrac{n}{3}\right) + n & &= 3\left(3T\left(\tfrac{n}{3^2}\right) + \tfrac{n}{3}\right) + n \\
&= 3^2T\left(\tfrac{n}{3^2}\right) + 2n & &= 3^2\left(3T\left(\tfrac{n}{3^3}\right) + \tfrac{n}{3^2}\right) + 2n \\
&= 3^3T\left(\tfrac{n}{3^3}\right) + 3n \\
&\ \ \vdots \qquad \vdots \\
&= 3^iT\left(\tfrac{n}{3^i}\right) + in \\
&\ \ \vdots \qquad \vdots \\
&= 3^{\log_3 n}T\left(\tfrac{n}{3^{\log_3 n}}\right) + n\log_3 n & &= n + n\log_3 n
\end{aligned}$$

Iterating Recurrences: Example 5

Consider

$$T(n) = \begin{cases} 1, & \text{if } n = 1, \\ 4T(n/2) + n, & \text{if } n \geq 2. \end{cases}$$

$$\begin{aligned}
T(n) &= 4T\left(\tfrac{n}{2}\right) + n & &= 4\left(4T\left(\tfrac{n}{2^2}\right) + \tfrac{n}{2}\right) + n \\
&= 4^2 T\left(\tfrac{n}{2^2}\right) + \tfrac{4}{2}n + n & &= 4^2\left(4T\left(\tfrac{n}{2^3}\right) + \tfrac{n}{2^2}\right) + \tfrac{4}{2}n + n \\
&= 4^3 T\left(\tfrac{n}{2^3}\right) + \tfrac{4^2}{2^2}n + \tfrac{4}{2}n + n \\
&\ \ \vdots \qquad \vdots \\
&= 4^i T\left(\tfrac{n}{2^i}\right) + \tfrac{4^{i-1}}{2^{i-1}}n + \qquad \tfrac{4^2}{2^2}n + n \\
&\ \ \vdots \qquad \vdots \\
&= 4^{\log_2 n} T\left(\tfrac{n}{2^{\log_2 n}}\right) + \tfrac{4^{\log_2 n - 1}}{2^{\log_2 n - 1}}n + \cdots + \tfrac{4}{2}n + n \\
&= 2n^2 - n
\end{aligned}$$

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences (to obtain the closed-form)
- Three different behaviors

Three Different Behaviors

Compare the iteration for the recurrences

- $T(n) = 2T(n/2) + n$ $\quad nT(1) + n\log_2 n$
- $T(n) = T(n/2) + n$ $\quad \Theta(n)$
- $T(n) = 4T(n/2) + n$ $\quad 2n^2 - n$

Anything in common?

In each case, size of subproblem in next iteration is half the size in the preceding iteration level.

All three recurrences iterate $\log_2 n$ times.

Three Different Behaviors

Theorem: Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative. Then we have the following big Θ bounds on the solution:

- If $a < 2$, then $T(n) = \Theta(n)$.
- If $a = 2$, then $T(n) = \Theta(n \log n)$.
- If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$.

We will now prove the case with $a > 2$.

Proof

$T(n) = aT(n/2) + n$, where $a > 2$. Assume that $n = 2^i$.

$$T(n) = a^i T\left(\frac{n}{2^i}\right) + \left(\frac{a^{i-1}}{2^{i-1}} + \frac{a^{i-2}}{2^{i-2}} + \cdots \frac{a}{2} + 1\right) n$$

$$T(n) = a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$

Work at "bottom"

Iterated Work

Proof

$$a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$

$$\Theta\left(n^{\log_2 a}\right) \qquad \Theta\left(n^{\log_2 a}\right)$$

Since $a > 2$, the geometric series is Θ of the largest term.

$$n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i = n \frac{1-(a/2)^{\log_2 n}}{1-a/2} = n\Theta((a/2)^{\log_2 n - 1})$$

To obtain $n(a/2)^{\log_2 n - 1}$:

$$n\left(\frac{a}{2}\right)^{\log_2 n - 1} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{2^{\log_2 n}} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{n} = \frac{2}{a} \cdot a^{\log_2 n}$$

Note that

$$a^{\log_2 n} = \left(2^{\log_2 a}\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^{\log_2 a} = n^{\log_2 a}$$

Example 5 Recap

Theorem: Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative.

- ...
- If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$.

Example:

$$T(n) = \begin{cases} 1, & \text{if } n = 1, \\ 4T(n/2) + n, & \text{if } n \geq 2. \end{cases}$$

$a = 4$, so the Theorem says that

$$T(n) = \Theta\left(n^{\log_2 a}\right) = \Theta\left(n^{\log_2 4}\right) = \Theta(n^2)$$

This matches with the exact answer of $2n^2 - n$.

Three Different Behaviors

Theorem: Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative. Then we have the following big Θ bounds on the solution:

- If $a < 2$, then $T(n) = \Theta(n)$.
- If $a = 2$, then $T(n) = \Theta(n \log n)$.
- If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$.

The Master Theorem

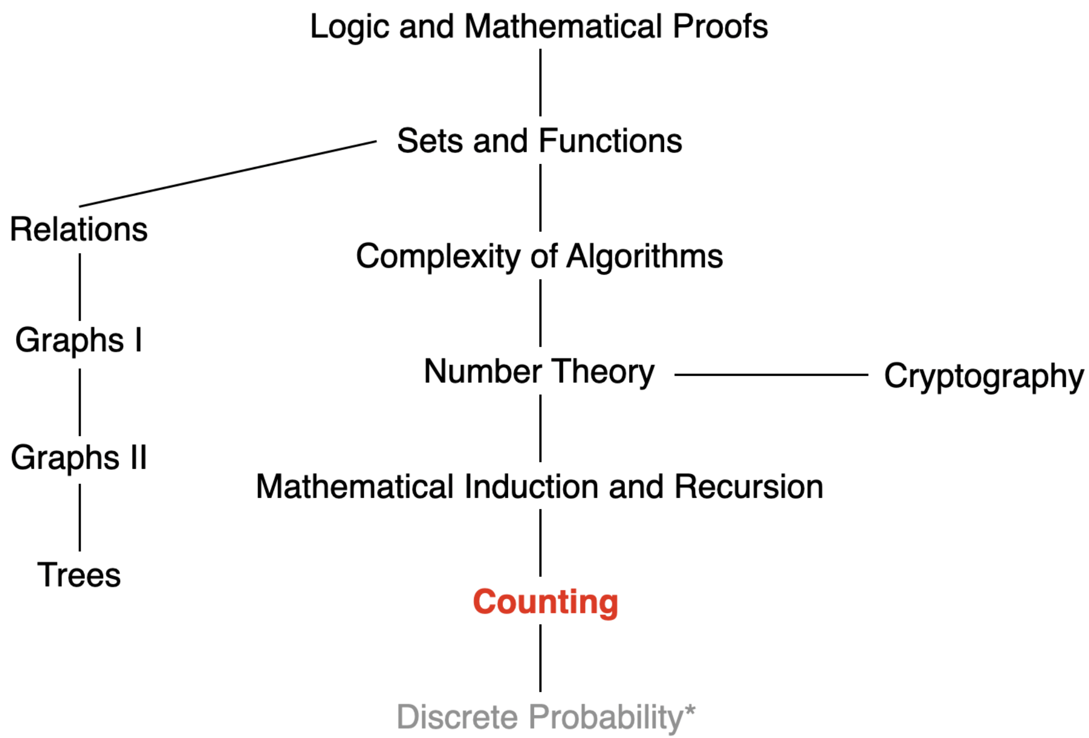
Theorem: Suppose that we have a recurrence of the form

$$T(n) = aT(n/b) + cn^d,$$

where a is a positive integer, $b \geq 1$, c, d are real numbers with c positive and d nonnegative, and $T(1)$ is nonnegative. Then we have the following big Θ bounds on the solution:

- If $a < b^d$, then $T(n) = \Theta(n^d)$.
- If $a = b^d$, then $T(n) = \Theta(n^d \log n)$.
- If $a > b^d$, then $T(n) = \Theta(n^{\log_b a})$.

This Lecture



Counting basis, Permutations, ...

Counting

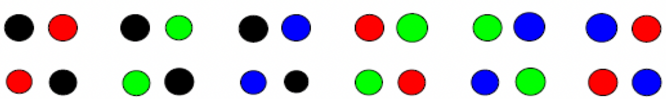
Assume we have a set of objects with certain properties

Counting is used to determine the number of these objects.

Example: How many different ways are there to choose 2 balls from

Unordered count?

Order counts?



Counting

Assume we have a set of objects with certain properties

Counting is used to determine the number of these objects.

Example:

- the number of steps in a computer program
- the number of passwords between 6 - 10 characters
- the number of telephone numbers with 8 digits

Counting may be very hard, not trivial.

Simplify the solution by decomposing the problem.

Basic Counting Rules

Product Rule:

- A count decomposes into a sequence of dependent counts.
- Each element in the first count is associated with all elements of the second count.

Sum Rule:

- A count decomposes into a set of independent counts.
- Elements of counts are alternatives.

The Product Rule

A count decomposes into a sequence of dependent counts.

The Product Rule: Suppose that a procedure can be broken down into a sequence of two tasks:

- There are n_1 ways to do the first task.
- For each of these ways of doing the first task, there are n_2 ways to do the second task.
- Then, there are $n_1 n_2$ ways to do the procedure.

Example: In an auditorium, the seats are labeled by a letter and numbers in between 1 to 50 (e.g., A23). What is the total number of seats?
$26 \times 50 = 1300$

The Product Rule

The Product Rule: If a count of elements can be broken down into a sequence of dependent counts where the first count yields n_1 elements, the second n_2 elements, and k-th count n_k elements, then the total number of elements is

$$n = n_1 \times n_2 \times ... \times n_k$$

Example:

- How many different bit strings of length seven are there? 2^7
- How many different functions are there from a set with m elements to a set with n elements? n^m
- How many one-to-one functions are there from a set with m elements to a set with n elements? $n(n-1)(n-2)...(n-m+1)$

The Product Rule: Example 1

What is the value of k after the following code, where n_1, n_2, . . ., n_m are positive integers, has been executed?

$$
\begin{array}{l}
k := 0 \\
\textbf{for } i_1 := 1 \textbf{ to } n_1 \\
\quad \textbf{for } i_2 := 1 \textbf{ to } n_2 \\
\qquad\qquad \cdot \\
\qquad\qquad \cdot \\
\qquad\qquad \cdot \\
\qquad \textbf{for } i_m := 1 \textbf{ to } n_m \\
\qquad\qquad k := k + 1
\end{array}
$$

$k = n_1 n_2 \ldots n_m$

The Product Rule: Example 2

If A_1, A_2, . . . , A_m are finite sets, then what is the number of elements in the Cartesian product of these sets?

$$|A_1 \times A_2 \times ... \times A_m| = |A_1||A2|...|Am|.$$

The Sum Rule

A count decomposes into a set of independent counts.

The Sum Rule:

- A task can be done either in one of n_1 ways or in one of n_2 ways
- None of the set of n_1 ways is the same as any of the set of n_2 ways

Then, there are $n_1 + n_2$ ways to do the task.

Example: You need to travel from city A to B. You may either fly, take a train, or a bus. There are 12 different flights, 5 different trains and 10 buses. How many options do you have to get from A to B? $12 + 5 + 10$

The Sum Rule

The Sum Rule: If a count of elements can be broken down into a set of independent counts where the first count yields n_1 elements, the second n_2 elements, and k-th count n_k elements, then the total number of elements is

$$n = n_1 + n_2 + ... + n_k.$$

The Sum Rule: Example 1

What is the value of k after the following code, where $n_1, n_2, \ldots, n_m$ are positive integers, has been executed?

$$
\begin{array}{l}
k := 0 \\
\textbf{for } i_1 := 1 \textbf{ to } n_1 \\
\quad k := k + 1 \\
\textbf{for } i_2 := 1 \textbf{ to } n_2 \\
\quad k := k + 1 \\
\qquad \vdots \\
\textbf{for } i_m := 1 \textbf{ to } n_m \\
\quad k := k + 1
\end{array}
$$

$k = n_1 + n_2 + \ldots + n_m$.

The Sum Rule: Example 2

If A_1, A_2, . . . , A_m are pairwise disjoint finite sets, then what is the number of elements in the union of these sets?

$$|A_1 \cup A_2 \cup \ldots \cup A_m| = |A_1| + |A_2| + \ldots + |A_m|.$$

More Complex Counting

Typically requires a combination of the sum and product rules.

Example: Each password is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution: Let P be the total number of possible passwords:

$$P = P_6 + P_7 + P_8.$$

Use P_6 as an example:

$$P_6 = (10+26)^6 - (26)^6 = 1,867,866,560.$$

The Subtraction Rule

The Subtraction Rule:

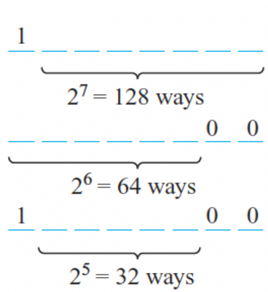
- A task can be done in either n_1 ways or n_2 ways

Then, the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common.

Principle of inclusion–exclusion:

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Example: How many bit strings of length eight either start with a 1 bit or end with the two bits 00 (inclusive or)?



$2^7 + 2^6 - 2^5$

The Division Rule

The Division Rule: There are n/d ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w, exactly d of the n ways correspond to way w.

Or equivalently, if f is a function from A to B, where A and B are finite sets, and that for every value $y \in B$ there are exactly d values $x \in A$ such that $f(x) = y$ (in which case, we say that f is d-to-one), then $|B| = |A|/d$.

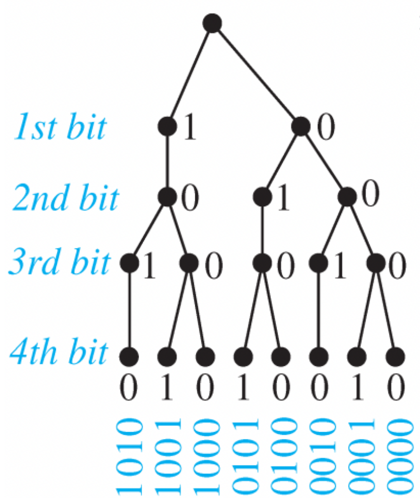
Example: How many different ways are there to seat four people around a circular table?
$4!/4 = 6$.

Tree Diagrams

A tree is a structure that consists of a root, branches and leaves.

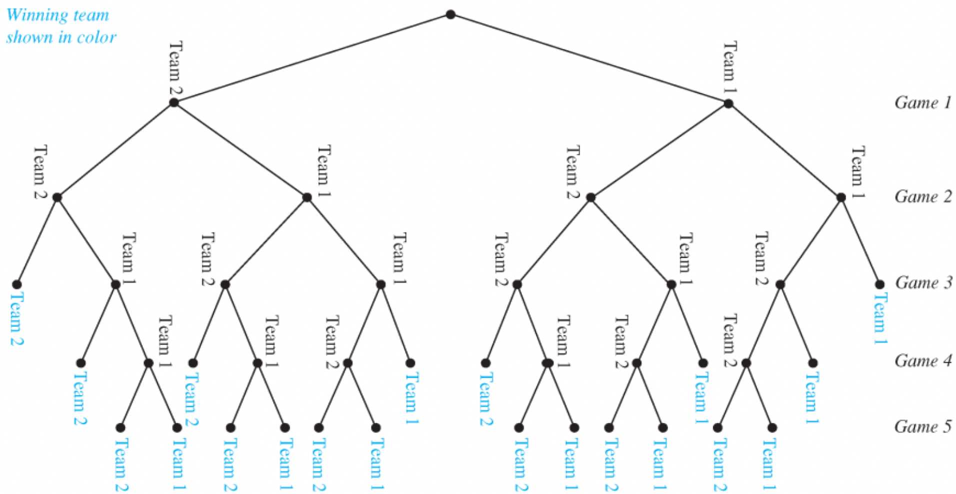
Can be useful to represent a counting problem and record the choices we made for alternatives. The count appears on the leaves.

Example: What is the number of bit strings of length 4 that do not have two consecutive 1's?



Tree Diagrams

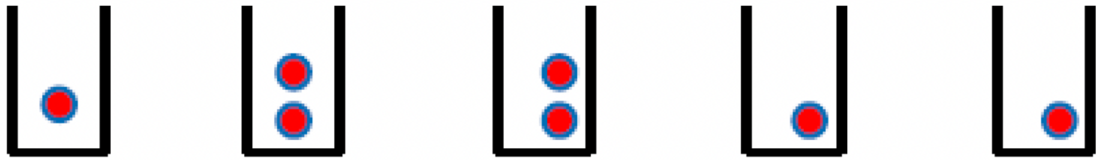
A playoff between two teams consists of at most five games. The first team that wins three games wins the playoff. In how many different ways can the playoff occur?



Pigeonhole Principle

Assume that there are a set of objects and a set of bins to store them.

The Pigeonhole Principle: If k is a positive integer and $k+1$ or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.



Pigeonhole Principle

The Pigeonhole Principle: If k is a positive integer and $k+1$ or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.

Proof by Contradiction: Suppose that none of the k boxes contains more than one object. Then the total number of objects would be at most k. This is a contradiction, because there are at least $k+1$ objects.

Example:

Assume that there are 367 students. Are there any two people who have the same birthday?

Pigeonhole Principle: Example

Show that for every integer n, there is a multiple of n that has only 0s and 1s in its decimal expansion.

Proof: Let n be a positive integer. Consider the $n+1$ integers

$$1, 11, 111, ..., 11...1,$$

where the last integer has $n+1$ 1s in its decimal expansion.

Note that there are n possible remainders when an integer is divided by n.

Because there are $n+1$ integers in this list, by the pigeonhole principle there must be two with the same remainder when divided by n.

The larger of these integers minus the smaller one is a multiple of n, which has a decimal expansion consisting entirely of 0s and 1s.

Generalized Pigeonhole Principle

There are 5 bins and 12 objects. Then there must be a bin with at least 3 objects. Why?

If N objects are placed into k bins, then there is at least one bin containing at least $\lceil N/k \rceil$ objects.

Example: Assume there are 100 students. How many of them were born in the same month?

Generalized Pigeonhole Principle

If N objects are placed into k bins, then there is at least one bin containing at least $\lceil N/k \rceil$ objects.

Proof: Suppose that none of the boxes contains more than $\lceil N/k \rceil$ objects. Then, the total number of objects is at most

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N$$

This is a contradiction because there are a total of N objects.

Pigeonhole Principle: Example 1

During a month with 30 days, a baseball team plays at least one game a day, but no more than 45 games.

Show that there must be a period of some number of consecutive days during which the team must play exactly 14 games.

Solution: Let a_j be the number of games played on or before the j th day of the month. Then,

$$a_1, a_2, \ldots, a_{30},$$

which is an increasing sequence of distinct integers, with $1 \leq a_j \leq 45$.

Moreover, $a_1 + 14, a_2 + 14, \ldots, a_{30} + 14$ is also an increasing sequence of distinct integers, with $15 \leq a_j + 14 \leq 59$.

Pigeonhole Principle: Example 1

During a month with 30 days, a baseball team plays at least one game a day, but no more than 45 games.

Show that there must be a period of some number of consecutive days during which the team must play exactly 14 games.

Solution: The 60 integers $a_1, a_2, ..., a_{30}, a_1 + 14, a_2 + 14, ..., a_{30} + 14$ are all less than or equal to 59. By the pigeonhole principle, two of these integers are equal.

Since the integers in each sequence are distinct, there must be indices i and j with $a_i = a_j + 14$.

Pigeonhole Principle: Example 2

Theorem: Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

Suppose that $a_1, a_2, \ldots, a_N$ is a sequence of real numbers:

- A subsequence of this sequence is a sequence of the form $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$, where $1 \leq i_1 < i_2 < \ldots < i_m \leq N$.
- A sequence is called strictly increasing if each term is larger than the one that precedes it.

Pigeonhole Principle: Example 2

Theorem: Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

Example: The sequence 8, 11, 9, 1, 4, 6, 12, 10, 5, 7 contains 10 terms. Note that $10 = 3^2 + 1$.

There are four strictly increasing subsequences of length four:

$$1, 4, 6, 12 \qquad 1, 4, 6, 7$$

$$1, 4, 6, 10 \qquad 1, 4, 5, 7$$

There is also a strictly decreasing subsequence of length four:

$$11, 9, 6, 5$$

Pigeonhole Principle: Example 2

Theorem: Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

Proof: Let $a_1, a_2, \ldots, a_{n^2+1}$ be a sequence of $n^2 + 1$ distinct real numbers. Associate (i_k, d_k) to the term a_k:

- i_k: the length of the longest increasing subsequence starting at a_k
- d_k: the length of the longest decreasing subsequence starting at a_k.

Suppose that there are no increasing or decreasing subsequences of length $n + 1$. I.e., $i_k \le n$ and $d_k \le n$ for $k = 1, 2, \ldots, n^2 + 1$.

By the product rule there are n^2 possible ordered pairs for (i_k, d_k). By the pigeonhole principle, two of these $n^2 + 1$ ordered pairs are equal.

That is, there exist terms a_s and a_t with $s < t$ such that $i_s = i_t$ and $d_s = d_t$.

Pigeonhole Principle: Example 2

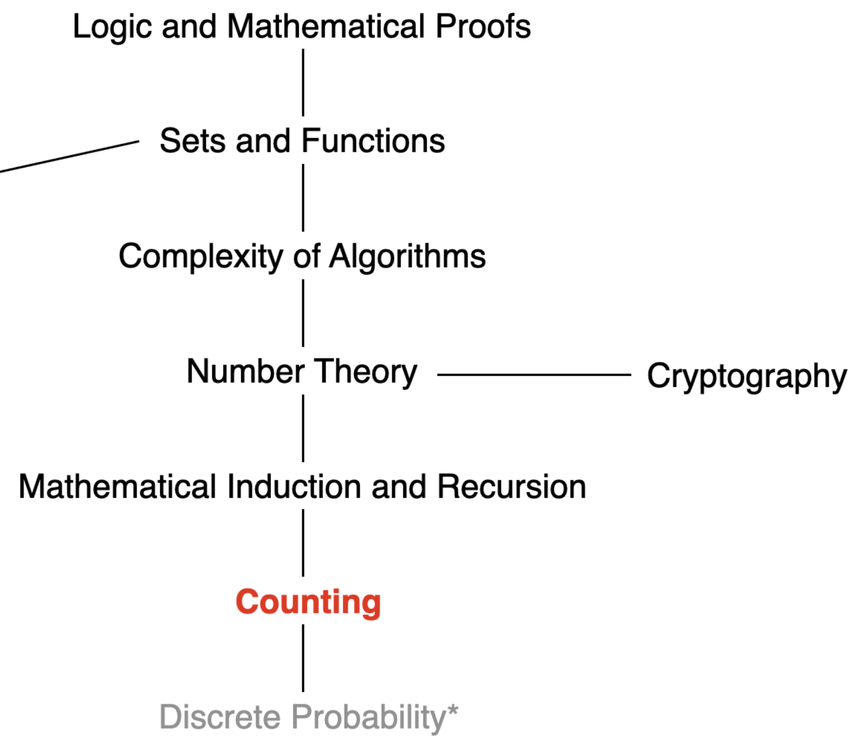
Theorem: Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

Proof: There exist terms a_s and a_t with $s < t$ such that $i_s = i_t$ and $d_s = d_t$. We will show that this is impossible.

The terms of the sequence are distinct, either $a_s < a_t$ or $a_s > a_t$:

- $a_s < a_t$: Since $i_s = i_t$, an increasing subsequence of length $i_t + 1$ can be built, i.e., a_s, a_t, ... (followed by an increasing subsequence of length i_t beginning at a_t)
- $a_s > a_t$, Since $d_s = d_t$, an decreasing sequence of length $d_t + 1$ can be built, i.e., a_s, a_t, ...

This Lecture



Counting basis, Permutations, ...

Permutations

A **permutation** of a set of distinct objects is an ordered arrangement of these objects.

An ordered arrangement of r elements of a set is called an **r-permutation**.

Example: Let $S = \{a, b, c\}$. The 2-permutations of S are the ordered arrangements (a, b), (a, c), (b, a) (b, c), (c, a). (c, b).

Permutations

Theorem: If n is a positive integer and r is an integer with $1 \leq r \leq n$, then there are

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1)$$

r-permutations of a set with n distinct elements.

Proof by the Product Rule: The first element of the permutation can be chosen in n ways, because there are n elements in the set.

There are $n-1$ ways to choose the second element of the permutation.

...

Permutations

Theorem: If n is a positive integer and r is an integer with $1 \leq r \leq n$, then there are

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1)$$

r-permutations of a set with n distinct elements.

Corollary: If n and r are integers with $0 \leq r \leq n$, then

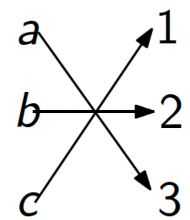
$$P(n, r) = \frac{n!}{(n-r)!}.$$

Bijections and Permutations

A function that is both one-to-one and onto is called a bijection, or a one-to-one correspondence.

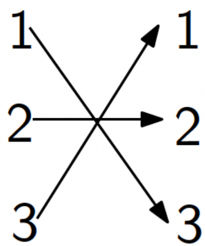
How many bijections are there?

$f : \{a, b, c\} \to \{1, 2, 3\}$ defined by $f(a) = 3, f(b) = 2, f(c) = 1$ is a bijection.



A bijection from a set onto itself is called a permutation.

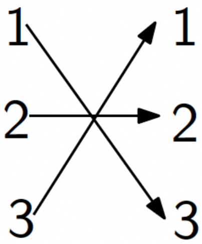
$f : \{1, 2, 3\} \to \{1, 2, 3\}$ defined by $f(1) = 3, f(2) = 2, f(3) = 1$ is a bijection.



Bijections and Permutations

A function that is both one-to-one and onto is called a bijection, or a one-to-one correspondence.
A bijection from a set onto itself is called a permutation.

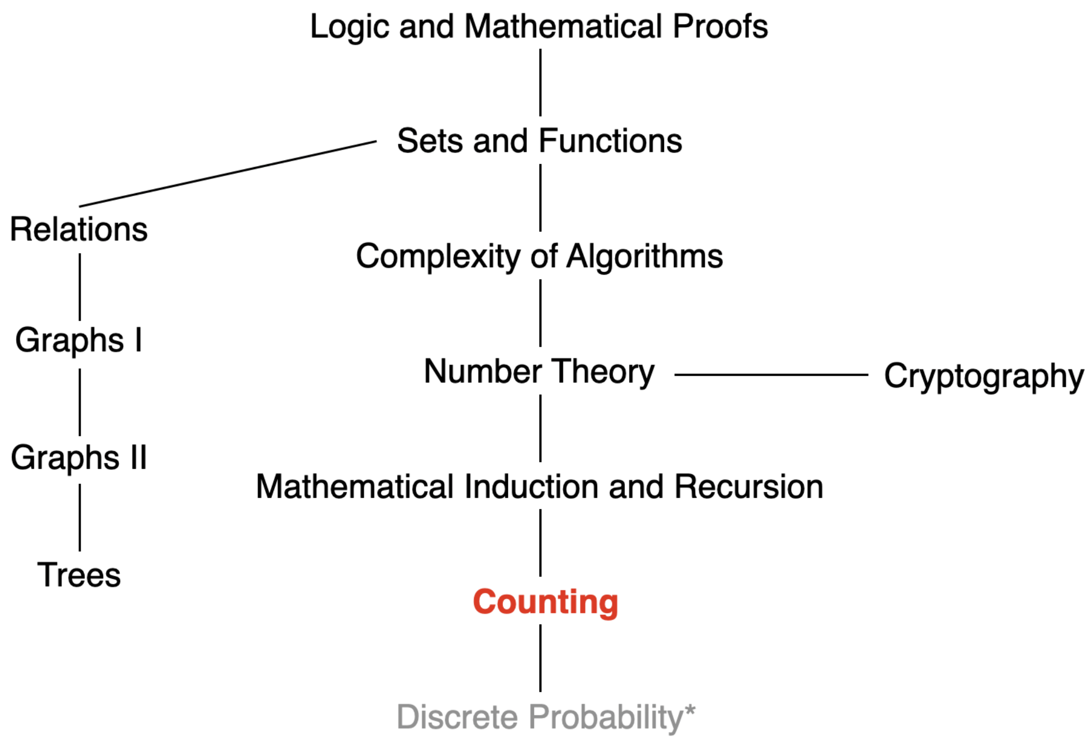
$f : \{1, 2, 3\} \rightarrow \{1, 2, 3\}$ defined by $f(1) = 3, f(2) = 2, f(3) = 1$ is a bijection.



In a bijection, exactly one arrow leaves each item on the left and exactly one arrow arrives at each item on the right.

Thus, the left and right sides must have the same size.

This Lecture



Counting basis, Permutations, ...