Discrete Mathematics for Computer Science

Lecture 11: Recursion and Counting

Dr. Ming Tang

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

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

Overview

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

Growth Rates of Solutions to Recurrences

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

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

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

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

1 32 48 64

1 32 48 64

Is $x > 32$?

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.

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: $T(n) = T(n/2) + 1$

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences: how to solve

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

- 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, by

- solving 2 subproblems of size $n/2$ and
- doing n units of additional work

or using $T(1)$ work for "bottom" case of $n = 1$

Iterating Recurrences: Example 1

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

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

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

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

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. Practically, we never explicitly perform this step, since it is obvious how the induction would work.

Iterating Recurrences: Example 2

Consider

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

$T(n) = T\left(\frac{n}{2}\right) + 1$

$T(n) = T\left(\frac{n}{2}\right) + 1 \qquad = \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1$

SUSTech Southern University of Science and Technology

Iterating Recurrences: Example 3

Consider

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

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

Iterating Recurrences: Example 4

Consider

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

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

Iterating Recurrences: Example 5

Consider

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

$$\begin{aligned}
T(n) &= 4T\left(\frac{n}{2}\right) + n \qquad = 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\
&= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n \quad = 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \\
&= 4^3 T\left(\frac{n}{2^3}\right) + \frac{4^2}{2^2}n + \frac{4}{2}n + n \\
&\quad \vdots \qquad \vdots \\
&= 4^i T\left(\frac{n}{2^i}\right) + \frac{4^{i-1}}{2^{i-1}}n + \qquad \frac{\,^2}{\,_2}n + n \\
&\quad \vdots \qquad \vdots \\
&= 4^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{4^{\log_2 n - 1}}{2^{\log_2 n - 1}}n + \cdots + \frac{4}{2}n + n \\
&= 2n^2 - n
\end{aligned}$$

Growth Rates of Solutions to Recurrences

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

Three Different Behaviors

Compare the iteration for the recurrences

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

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 $T(n) = \Theta(n \log_2 n)$.
- 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

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

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

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

Proof

n times the largest term in the geometric series is

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

Notice that

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

So the total work is

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

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

Example 5 Recap

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

$a = 4$, 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 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. Then we have the following big Θ bounds on the solution:

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

The Master Theorem

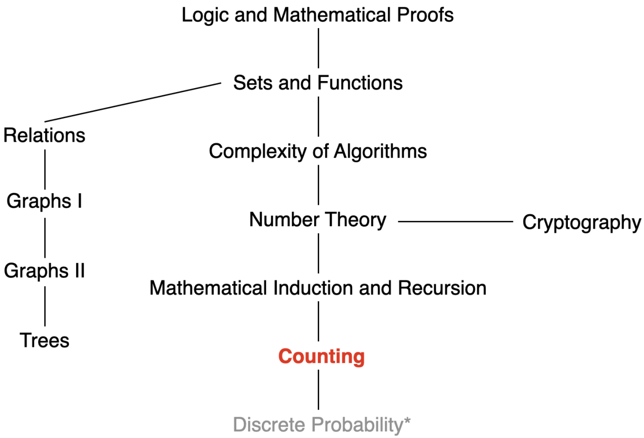
***Theorem:** Suppose that we have a recurrence of the form

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

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

This Lecture



Counting basis, Permutations, ...

Counting

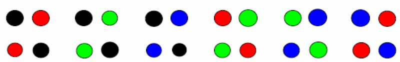
Assume we have a set of objects with certain properties

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 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?

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

$k = n_1 n_2 \ldots n_m$

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?

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

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

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

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?

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

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

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?

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

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:

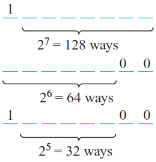
- A task can be done in either n_1 ways or n_2 ways

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:

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

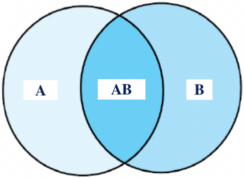
Example: How many bit strings of length eight either start with a 1 bit or end with the two bits 00 (inclusive or)?



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

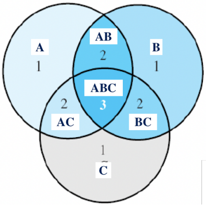
Inclusion-Exclusion Principle

Two sets A and B: $|A \cup B| = |A| + |B| - |A \cap B|$



Three sets A, B, and C:

$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$



Inclusion-Exclusion Principle

Let $E_1, E_2, \ldots, E_n$ be finite sets:

$$|E_1 \cup E_2 \cup \cdots \cup E_n| = \sum_{1 \le i \le n} |E_i| - \sum_{1 \le i < j \le n} |E_i \cap E_j| + \sum_{1 \le i < j < k \le n} |E_i \cap E_j \cap E_k| - \ldots + (-1)^{n+1} |E_1 \cap E_2 \cap \ldots \cap E_n|.$$

Or equivalently,

$$|\cup_{i=1}^{n} E_i| = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} |E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_k}|$$

Proof by Induction.

An Alternative Form of Inclusion–Exclusion Principle

This form can be used to solve problems that ask for the number of elements in a set that have none of n properties P_1, P_2, . . . , P_n.

- A_i: the subset containing the elements that have property P_i.
- $N(P_{i_1}, P_{i_2}, ..., P_{i_k})$: The number of elements with all the properties.

$$|A_{i_1} \cap A_{i_2} \cap ... \cap A_{i_k}| = N(P_{i_1}, P_{i_2}, ..., P_{i_k}).$$

- $N(P'_{i_1}, P'_{i_2}, ..., P'_n)$: The number of elements with none of the properties P_1, P_2, ... , P_n.

$$N(P'_{i_1}, P'_{i_2}, ..., P'_n) = N - |A_1 \cup A_2 \cup ... \cup A_n|.$$

$$N(P'_1 P'_2 \dots P'_n) = N - \sum_{1 \le i \le n} N(P_i) + \sum_{1 \le i < j \le n} N(P_i P_j) - \sum_{1 \le i < j < k \le n} N(P_i P_j P_k) + \cdots + (-1)^n N(P_1 P_2 \dots P_n).$$

Inclusion-Exclusion Principle: Example

How many onto functions are there from a set with six elements to a set with three elements?

Solution: Suppose that the elements in the codomain are b_1, b_2, and b_3.

Let P_1, P_2, and P_3 be the properties that b_1, b_2, and b_3 are not in the range of the function, respectively.

Let A_1, A_2, A_3 be the corresponding subsets of functions.

$$N(P_1', P_2', ..., P_3') = N - |A_1 \cup A_2 \cup A_3|$$

$$N(P_1'P_2'P_3') = N - [N(P_1) + N(P_2) + N(P_3)] \\ + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3),$$

Inclusion-Exclusion Principle: Example

How many onto functions are there from a set with six elements to a set with three elements?

$$N(P_1'P_2'P_3') = N - [N(P_1) + N(P_2) + N(P_3)] + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3),$$

- N: the total number of functions. $N = 3^6$
- $N(P_i)$: the number of functions that do not have b_i in their range. $N(P_i) = 2^6$
- $N(P_i, P_j)$: The number of functions that do not have b_i and b_j in their range. $N(P_i, P_j) = 1^6$
- $N(P_1, P_2, P_3) = 0$

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 $|B| = |A|/d$.

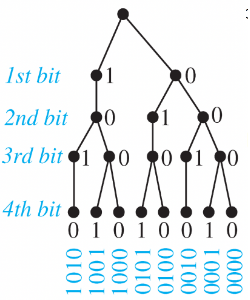
Example: How many different ways are there to seat four people around a circular table?
$4!/4 = 6$.

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.

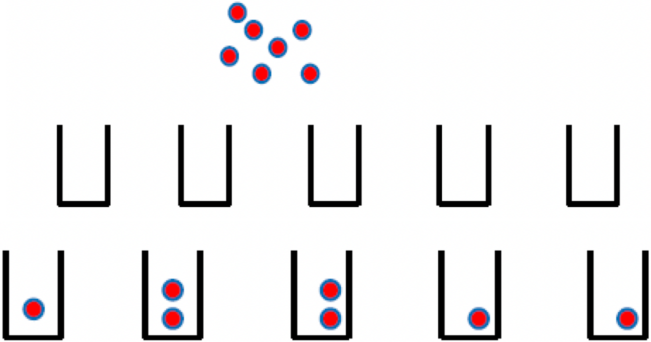
Example: What is the number of bit strings of length 4 that do not have two consecutive 1's?



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

$$1, 11, 111, \ldots, 11\ldots1,$$

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,

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

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

Moreover, $a_1 + 14, a_2 + 14, \dots, a_{30} + 14$ 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., $i_k \leq n$ and $d_k \leq n$ for $k = 1, 2, ..., n^2 + 1$.

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

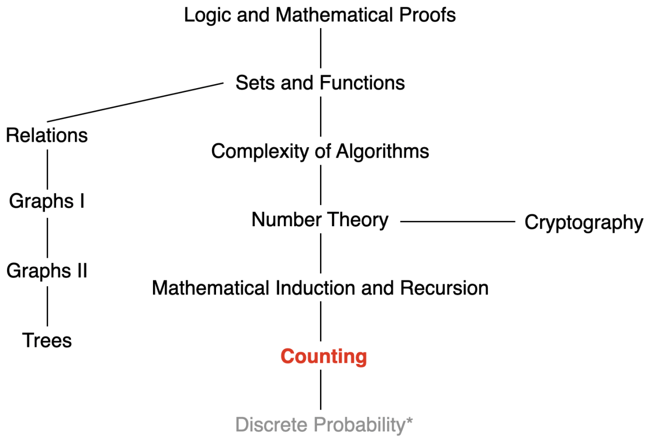
Theorem: Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

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 $a_s > a_t$:

- $a_s < a_t$: 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); thus, $i_s > i_t$
- $a_s > a_t$: $d_s = d_t$; an decreasing sequence of length $d_t + 1$ can be built, i.e., a_s, a_t, ...; thus, $d_s > d_t$;

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

How many 3-permutations of $\{1, 2, ..., n\}$ are there?

Based on product rule:

- n choices for first number.
- For each way of choosing first number, there are $n - 1$ choices for the second.
- For each way of choosing first two numbers, there are $n - 2$ choices for the third number.

By product rule, there are $n(n - 1)(n - 2)$ ways to choose 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.

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

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

Permutations: Example

Example 1: How many ways are there to select a first-prize winner, a second-prize winner, and a third-prize winner from 100 different people who have entered a contest?

$P(100, 3) = 100 \times 99 \times 98 = 970,200.$

Example 2: How many permutations of the letters ABCDEFGH contain the string ABC?

The letters ABC must occur as a block. Thus, it is equivalent to finding the number of permutations of six objects:

$$\text{ABC, D, E, F, G, H.}$$

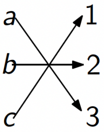
Thus, there are $P(6, 6) = 6! = 720$ permutations.

Bijections and Permutations

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

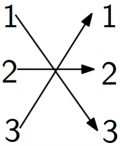
How many bijections are there?

$f : \{a, b, c\} \rightarrow \{1, 2, 3\}$ 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.

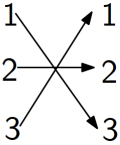
$f : \{1, 2, 3\} \rightarrow \{1, 2, 3\}$ defined by $f(1) = 3, f(2) = 2, f(3) = 1$ is a bijection.



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.

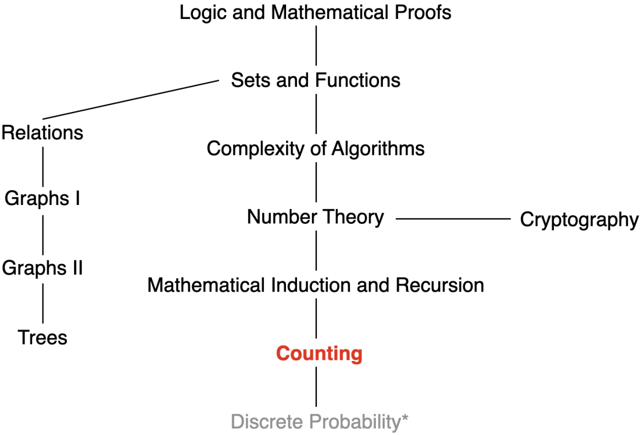
$f : \{1, 2, 3\} \to \{1, 2, 3\}$ defined by $f(1) = 3, f(2) = 2, f(3) = 1$ is a bijection.



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, Combinations, ...