Discrete Mathematics

Richard Anderson
University of Washington

Today's topics

- Teaching Discrete Mathematics
- Active Learning in Discrete Mathematics
- Educational Technology Research at UW
- Big Ideas: Complexity Theory

Highlights from Day 1

Website

- http://cs.washington.edu/homes/anderson
  - Home page
- http://cs.washington.edu/homes/anderson/iucee
  - Workshop websites
  - Updates might be slow (through July 20)
- Google groups
  - IUCEE Workshop on Teaching Algorithms

Re-revised Workshop Schedule

- Monday, June 30, Active learning and instructional goals
  - Morning
    - Welcome and Overview (1 hr)
    - Introductory Activity (1 hr). Determine background of participants
    - Active learning and instructional goals (1hr) in Discrete Math, Data Structures, Algorithms.
  - Afternoon
    - Group Work (1.5 hrs). Development of activities/goals from participant's classes.
    - Content lectures (Great Ideas in Computing): (1.5 hr) Problem mapping
- Tuesday, July 1, Discrete Mathematics
  - Morning
    - Discrete Mathematics Teaching (2 hrs)
    - Activities in Discrete Mathematics (1 hr)
  - Afternoon
    - Educational Technology Lecture (1.5 hrs)
    - Content Lecture: (1.5 hrs) Complexity Theory
- Wednesday, July 2, Data Structures
  - Morning
    - Data Structures Teaching (2hrs)
    - Data Structure Activities (1 hr)
  - Afternoon
    - Group work (1.5 hrs)
    - Content Lecture: (1.5 hr) Average Case Analysis
- Thursday, July 3, Algorithms
  - Morning
    - Algorithms Teaching (2 hrs)
    - Algorithms Activities (1 hr)
  - Afternoon
    - Activity Critique (.5 hr)
    - Research discussion (1 hr)
    - Theory discussion (optional)
- Friday, July 4, Topics
  - Morning
    - Content Lecture (1.5 hrs) Algorithm implementation
    - Lecture (1.5 hrs) Socially relevant computing
  - Afternoon
    - Follow up and faculty presentations (1.5 hrs)
    - Research Discussion (1.5 hrs)

Wednesday

- Each group:
  - Design two classroom activities for your classes. Identify the pedagogical goals of the activity.
- Five of the groups will give progress report to the class
- Overnight each group should prepare ppt slides
- Thursday there will be a feedback/critique session

Thursday and Friday

- Each group will develop a presentation on how they are going to apply ideas from this workshop.
- Thursday
  - Two hours work time
- Friday
  - Three hours presentation time
    - 15 minutes per group with PPT slides

University of Washington Course

CSE 321 Discrete Structures (4)
Fundamentals of set theory, graph theory, enumeration, and algebraic structures, with applications in computing. Prerequisite: CSE 143; either MATH 126, MATH 129, or MATH 136.

- Discrete Mathematics and Its Applications, Rosen, 6-th Edition
- Ten week term
  - 3 lectures per week (50 minutes)
  - 1 quiz section
  - Midterm, Final

Course overview

- Logic (4)
- Reasoning (2)
- Set Theory (1)
- Number Theory (4)
- Counting (3)
- Probability (3)
- Relations (3)
- Graph Theory (2)

Analyzing the course and content

- What is the purpose of each unit?
  - Long term impact on students
- What are the learning goals of each unit?
  - How are they evaluated
- What strategies can be used to make material relevant and interesting?
- How does the context impact the content

Broader goals

- Analysis of course content
  - How does this apply to the courses that you teach?
- Reflect on challenges of your courses

Overall course context

- First course in CSE Major
  - Students will have taken CS1, CS2
  - Various mathematics and physics classes
- Broad range of mathematical background of entering students
- Goals of the course
  - Formalism for later study
  - Learn how to do a mathematical argument
- Many students are not interested in this course

Logic

- Begin by motivating the entire course
  - "Why this stuff is important"
- Formal systems used throughout computing
- Propositional logic and predicate calculus
- Boolean logic covered multiple time in curriculum
- Relationship between logic and English is hard for the students
  - implication and quantification

Goals

- Understanding boolean algebra
- Connection with language
  - Represent statements with logic
- Predicates
  - Meaning of quantifiers
  - Nested quantification

Why this material is important

- Language and formalism for expressing ideas in computing
- Fundamental tasks in computing
  - Translating imprecise specification into a working system
  - Getting the details right

Propositions

- A statement that has a truth value
- Which of the following are propositions?
  - The Washington State flag is red
  - It snowed in Whistler, BC on January 4, 2008.
  - Hillary Clinton won the democratic caucus in Iowa
  - Space aliens landed in Roswell, New Mexico
  - Ron Paul would be a great president
  - Turn your homework in on Wednesday
  - Why are we taking this class?
  - If n is an integer greater than two, then the equation $a^n + b^n = c^n$ has no solutions in non-zero integers a, b, and c.
  - Every even integer greater than two can be written as the sum of two primes
  - This statement is false
- Propositional variables: *p, q, r, s, . . .*
- Truth values: **T** for true, **F** for false

Compound Propositions

- Negation (not) $\neg p$
- Conjunction (and) $p \wedge q$
- Disjunction (or) $p \vee q$
- Exclusive or $p \oplus q$
- Implication $p \rightarrow q$
- Biconditional $p \leftrightarrow q$

$p \rightarrow q$

p	q	$p \rightarrow q$

- Implication
  - *p* implies *q*
  - whenever *p* is true *q* must be true
  - if *p* then *q*
  - *q* if *p*
  - *p* is sufficient for *q*
  - *p* only if *q*

English and Logic

- You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old
  - q: you can ride the roller coaster
  - r: you are under 4 feet tall
  - s: you are older than 16

$(r \land \neg s) \to \neg q$

$\neg s \to (r \to \neg q)$

Logical equivalence

- Terminology: A compound proposition is a
  - Tautology if it is always true
  - Contradiction if it is always false
  - Contingency if it can be either true or false

$p \lor \neg p$

$(p \oplus p) \lor p$

$p \oplus \neg p \oplus q \oplus \neg q$

$(p \to q) \land p$

$(p \land q) \lor (p \land \neg q) \lor (\neg p \land q) \lor (\neg p \land \neg q)$

Logical Proofs

- To show P is equivalent to Q
  - Apply a series of logical equivalences to subexpressions to convert P to Q
- To show P is a tautology
  - Apply a series of logical equivalences to subexpressions to convert P to **T**

Statements with quantifiers

Domain: Positive Integers

Even(x)
Odd(x)
Prime(x)
Greater(x,y)
Equal(x,y)

- $\exists x$ Even(x)
- $\forall x$ Odd(x)
- $\forall x$ (Even(x) $\lor$ Odd(x))
- $\exists x$ (Even(x) $\land$ Odd(x))
- $\forall x$ Greater(x+1, x)
- $\exists x$ (Even(x) $\land$ Prime(x))

Statements with quantifiers

- $\forall x \exists y$ Greater (y, x)

  For every number there is some number that is greater than it
- $\exists$ y $\forall$ x Greater (y, x)
- $\forall x \exists y$ (Greater(y, x) $\land$ Prime(y))
- $\forall x$ (Prime(x) $\to$ (Equal(x, 2) $\lor$ Odd(x))
- $\exists x \exists y$(Equal(x, y + 2) $\land$ Prime(x) $\land$ Prime(y))

Domain: Positive Integers

Greater(a, b) ≡ "a > b"

Prolog

- Logic programming language
- Facts and Rules

```
RunsOS(SlipperPC,  Windows)
RunsOS(SlipperTablet, Windows)
RunsOS(CarmelLaptop, Linux)

OSVersion(SlipperPC, SP2)
OSVersion(SlipperTablet, SP1)
OSVersion(CarmelLaptop, Ver3)

LaterVersion(SP2, SP1)
LaterVersion(Ver3, Ver2)
LaterVersion(Ver2, Ver1)
```

```
Later(x, y) :-
   Later(x, z), Later(z, y)

NotLater(x, y) :- Later(y, x)
NotLater(x, y) :-
   SameVersion(x, y)

MachineVulnerable(m) :-
   OSVersion(m, v),
   VersionVulnerable(v)
VersionVulnerable(v) :-
   CriticalVulnerability(x),
   Version(x, n),
   NotLater(v, n)
```

Nested Quantifiers

- Iteration over multiple variables
- Nested loops
- Details
  - Use distinct variables
    - $\forall x(\exists y(P(x,y) \rightarrow \forall x\, Q(y, x)))$
  - Variable name doesn't matter
    - $\forall x \exists y\, P(x, y) \equiv \forall a \exists b\, P(a, b)$
  - Positions of quantifiers can change (but order is important)
    - $\forall x\, (Q(x) \wedge \exists y\, P(x, y)) \equiv \forall x \exists y\, (Q(x) \wedge P(x, y))$

Quantification with two variables

Expression	When true	When false
$\forall x \forall y\, P(x,y)$		
$\exists x \exists y\, P(x,y)$		
$\forall x \exists y\, P(x, y)$		
$\exists y \forall x\, P(x, y)$		

Reasoning

- Students have difficulty with mathematical proofs
- Attempt made to introduce proofs
- Describe proofs by technique
- Some students have difficulty appreciating a direct proof
- Proof by contradiction leads to confusion

Goals

- Understand the basic notion of a proof in a formal system
- Derive and recognize mathematically valid proofs
- Understand basic proof techniques

Reasoning

- "If Seattle won last Saturday they would be in the playoffs"
- "Seattle is not in the playoffs"
- Therefore . . .

Proofs

- Start with hypotheses and facts
- Use rules of inference to extend set of facts
- Result is proved when it is included in the set

Rules of Inference

$$\begin{array}{l} p \\ \underline{p \to q} \\ \therefore q \end{array} \qquad \begin{array}{l} \neg q \\ \underline{p \to q} \\ \therefore \neg p \end{array} \qquad \begin{array}{l} p \to q \\ \underline{q \to r} \\ \therefore p \to r \end{array} \qquad \begin{array}{l} p \vee q \\ \underline{\neg p} \\ \therefore q \end{array}$$

$$\begin{array}{l} \underline{p} \\ \therefore p \vee q \end{array} \qquad \begin{array}{l} p \\ \underline{q} \\ \therefore p \vee q \end{array} \qquad \begin{array}{l} \underline{p \wedge q} \\ \therefore p \end{array} \qquad \begin{array}{l} p \vee q \\ \underline{\neg p \vee r} \\ \therefore q \vee r \end{array}$$

$$\begin{array}{l} \underline{\forall x\, P(x)} \\ \therefore P(c) \end{array} \qquad \begin{array}{l} \underline{P(c) \text{ for any } c} \\ \therefore \forall x P(x) \end{array} \qquad \begin{array}{l} \underline{\exists x\, P(x)} \\ \therefore P(c) \text{ for some } c \end{array} \qquad \begin{array}{l} \underline{P(c) \text{ for some } c} \\ \therefore \exists x\, P(x) \end{array}$$

Proofs

- Proof methods
  - Direct proof
  - Contrapositive proof
  - Proof by contradiction
  - Proof by equivalence

Direct Proof

- If n is odd, then n^2 is odd

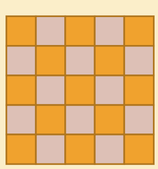
Definition
n is even if $n = 2k$ for some integer k
n is odd if $n = 2k+1$ for some integer k

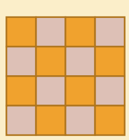
Contradiction example

- Show that at least four of any 22 days must fall on the same day of the week

Tiling problems

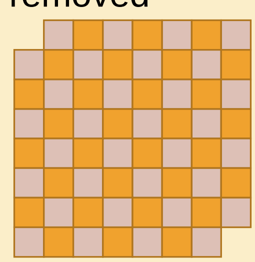
- Can an $n \times n$ checkerboard be tiled with 2×1 tiles?





8× 8 Checkerboard with two corners removed

- Can an 8×8 checkerboard with upper left and lower right corners removed be tiled with 2×1 tiles?



Set Theory

- Students have seen this many times already
- Still important for students to see the definitions / terminology
- Russell's Paradox discussed

Definition: A set is an unordered collection of objects

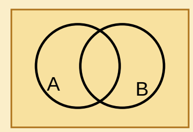
Cartesian Product : $A \times B$

$$A \times B = \{ (a, b) \mid a \in A \wedge b \in B\}$$

De Morgan's Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$



Proof technique:
To show C = D show
$x \in C \rightarrow x \in D$ and
$x \in D \rightarrow x \in C$

Russell's Paradox

$$S = \{ x \mid x \notin x \}$$

Number Theory

- Important for a small number of computing applications
  - Students should know a little number theory to appreciate aspects of security
- Students who will go on to graduate school should know this stuff
- Concepts such as modular arithmetic important for algorithmic thinking
- Mixed background of students coming in
  - Top students understand this from their math classes
  - Other students unable to transfer knowledge from other disciplines

Goals

- Understand modular arithmetic
- Provide motivating example
  - RSA encryption
  - Students should understand what public key cryptography is, but the details do not need to be retained
  - Something of interest for most advanced students
- Introduce algorithmic and computational topics
  - Fast exponentiation

Arithmetic mod 7

- $a +_7 b = (a + b) \bmod 7$
- $a \times_7 b = (a \times b) \bmod 7$

+	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

x	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

Multiplicative Inverses

- Euclid's theorem: if *x* and *y* are relatively prime, then there exists integers *s*, *t*, such that:

$$sx + ty = 1$$

- Prove $a \in \{1, 2, 3, 4, 5, 6\}$ has a multiplicative inverse under $\times_7$

Hashing

- Map values from a large domain, 0…M-1 in a much smaller domain, 0…n-1
- Index lookup
- Test for equality
- Hash(x) = x mod p
- Often want the hash function to depend on all of the bits of the data
  - Collision management

Pseudo Random number generation

- Linear Congruential method

$$x_{n+1} = (a\, x_n + c) \bmod m$$

Modular Exponentiation

x	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

a	a^1	a^2	a^3	a^4	a^5	a^6
1						
2						
3						
4						
5						
6						

Exponentiation

- Compute 78365^{81453}

- Compute 78365^{81453} mod 104729

Primality

- An integer p is prime if its only divisors are 1 and p
- An integer that is greater than 1, and not prime is called composite
- Fundamental theorem of arithmetic:
  - Every positive integer greater than one has a unique prime factorization

Distribution of Primes

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263
269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359

- If you pick a random number n in the range [x, 2x], what is the chance that n is prime?

Famous Algorithmic Problems

- Primality Testing:
  - Given an integer n, determine if n is prime
- Factoring
  - Given an integer n, determine the prime factorization of n

Primality Testing

- Is the following 200 digit number prime:

40992408416096028179761232532587525402909285099086220133403920525409552083528606215439915948260875718893797824735118621138192569490840098061133066650255608065609253901288801302035441884878187944219033

Public Key Cryptography

- How can Alice send a secret message to Bob if Bob cannot send a secret key to Alice?

My public key is:

13890580304018329082310291802198210923810830129823019128092183021398301292381320498068029809347849394598178479388287398457923893848928823748283829929384020010924380915809283290823823

RSA

- Rivest – Shamir – Adelman
- $n = pq$. p, q are large primes
- Choose e relatively prime to $(p-1)(q-1)$
- Find d, k such that $de + k(p-1)(q-1) = 1$ by Euclid's Algorithm
- Publish e as the encryption key, d is kept private as the decryption key

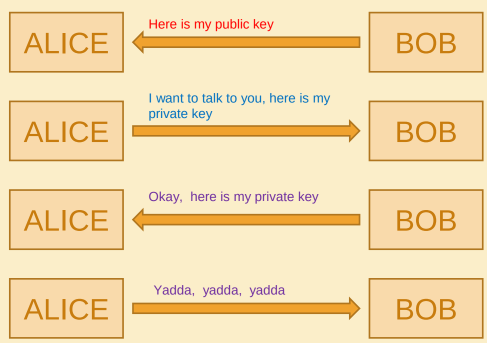
Message protocol

- Bob
  - Precompute p, q, n, e, d
  - Publish e, n
- Alice
  - Read e, n from Bob's public site
  - To send message M, compute $C = M^e \bmod n$
  - Send C to Bob
- Bob
  - Compute C^d to decode message M

Decryption

- $de = 1 + k(p-1)(q-1)$
- $C^d \equiv (M^e)^d = M^{de} = M^{1 + k(p-1)(q-1)} \pmod{n}$
- $C^d \equiv M\,(M^{p-1})^{k(q-1)} \equiv M \pmod{p}$
- $C^d \equiv M\,(M^{q-1})^{k(p-1)} \equiv M \pmod{q}$
- Hence $C^d \equiv M \pmod{pq}$

Practical Cryptography



Induction

- Considered to be most important part of the course
- Students will have seen basic induction
  - but more sophisticated uses are new
    - "Strong induction"
  - link it with formal proof
  - recursion is new to most students
- Matter of discussion how formal to make the coverage

Goals

- Be able to use induction in mathematical arguments
  - understand how to use induction hypothesis
- Give recursive definitions of sets, strings, and trees
- Be able to use structural induction to establish properties of recursively defined objects
- Appreciate that there is a formal structure underneath computational objects

Induction Example

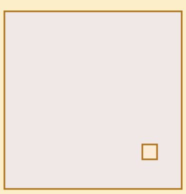
- Prove $3 \mid 2^{2n} - 1$ for $n \geq 0$

Induction as a rule of Inference

$$\begin{array}{l} P(0) \\ \underline{\forall k\, (P(k) \rightarrow P(k+1))} \\ \therefore \forall n\, P(n) \end{array}$$

Cute Application: Checkerboard Tiling with Trinominos

Prove that a $2^k \times 2^k$ checkerboard with one square removed can be tiled with: 



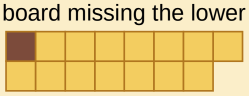
Strong Induction

$$\begin{array}{l} P(0) \\ \underline{\forall k\, ((P(0) \wedge P(1) \wedge P(2) \wedge \ldots \wedge P(k)) \rightarrow P(k+1))} \\ \therefore \forall n\, P(n) \end{array}$$

Player 1 wins n × 2 Chomp!

Winning strategy: chose the lower corner square

Theorem: Player 2 loses when faced with an $n \times 2$ board missing the lower corner square



Recursive Definitions

- $F(0) = 0$; $F(n + 1) = F(n) + 1$;
- $F(0) = 1$; $F(n + 1) = 2 \times F(n)$;
- $F(0) = 1$; $F(n + 1) = 2^{F(n)}$

Recursive Definitions of Sets

- Recursive definition
  - Basis step: $0 \in S$
  - Recursive step: if $x \in S$, then $x + 2 \in S$
  - Exclusion rule: Every element in S follows from basis steps and a finite number of recursive steps

Strings

- The set Σ^* of strings over the alphabet Σ is defined
  - Basis: $\lambda \in \Sigma^*$ (λ is the empty string)
  - Recursive: if $w \in \Sigma^*$, $x \in \Sigma$, then $wx \in \Sigma^*$

Families of strings over $\Sigma = \{a, b\}$

- L_1
  - $\lambda \in L_1$
  - $w \in L_1$ then $awb \in L_1$
- L_2
  - $\lambda \in L_2$
  - $w \in L_2$ then $aw \in L_2$
  - $w \in L_2$ then $wb \in L_2$

Function definitions

$Len(\lambda) = 0;$
$Len(wx) = 1 + Len(w)$; for $w \in \Sigma^*$, $x \in \Sigma$

$Concat(w, \lambda) = w$ for $w \in \Sigma^*$
$Concat(w_1,w_2x) = Concat(w_1,w_2)x$ for w_1, w_2 in Σ^*, $x \in \Sigma$

Tree definitions

- A single vertex r is a tree with root r.
- Let $t_1, t_2, \ldots, t_n$ be trees with roots $r_1, r_2, \ldots, r_n$ respectively, and let r be a vertex. A new tree with root r is formed by adding edges from r to $r_1, \ldots, r_n$.

Simplifying notation

- $(\bullet, T_1, T_2)$, tree with left subtree T_1 and right subtree T_2
- ε is the empty tree
- Extended Binary Trees (EBT)
  - $\varepsilon \in$ EBT
  - if $T_1, T_2 \in$ EBT, then $(\bullet, T_1, T_2) \in$ EBT
- Full Binary Trees (FBT)
  - $\bullet \in$ FBT
  - if $T_1, T_2 \in$ FBT, then $(\bullet, T_1, T_2) \in$ FBT

Recursive Functions on Trees

- N(T) - number of vertices of T
- $N(\varepsilon) = 0$; $N(\bullet) = 1$
- $N(\bullet, T_1, T_2) = 1 + N(T_1) + N(T_2)$

- Ht(T) – height of T
- $Ht(\varepsilon) = 0$; $Ht(\bullet) = 1$
- $Ht(\bullet, T_1, T_2) = 1 + \max(Ht(T_1), Ht(T_2))$

NOTE: Height definition differs from the text
Base case $H(\bullet) = 0$ used in text

Structural Induction

- Show P holds for all basis elements of S.
- Show that P holds for elements used to construct a new element of S, then P holds for the new elements.

Binary Trees

- If T is a binary tree, then $N(T) \le 2^{Ht(T)} - 1$

If $T = \varepsilon$:

If $T = (\bullet, T_1, T_2)$ $\quad Ht(T_1) = x$, $Ht(T_2) = y$
$N(T_1) \le 2^x$, $N(T_2) \le 2^y$

$N(T) = N(T_1) + N(T_2) + 1$
$\le 2^x - 1 + 2^y - 1 + 1$
$\le 2^{Ht(T) - 1} + 2^{Ht(T) - 1} - 1$
$\le 2^{Ht(T)} - 1$

Counting

- Convey general rules of counting
- Material has been seen in math classes – but the connection to Computing is important
- Don't want to spend too much time on this because it is specialized and won't be retained
- Combinatorial proofs can be very clever (but its not clear what students get out of them)
- Some of this material has little general application
- Easy topic to for creating homework and exam questions

Goals

- Convey general rules of counting
  - Cartesian product is important
- Link material they have seen in math classes to computing
- Strengthen algorithmic skills by solving counting problems
  - Decomposition
  - Mapping

Counting Rules

Product Rule: If there are n_1 choices for the first item and n_2 choices for the second item, then there are $n_1 n_2$ choices for the two items

Sum Rule: If there are n_1 choices of an element from S_1 and n_2 choices of an element from S_2 and $S_1 \cap S_2$ is empty, then there are $n_1 + n_2$ choices of an element from $S_1 \cup S_2$

Counting examples

License numbers have the form LLL DDD, how many different license numbers are available?

There are 38 students in a class, and 38 chairs, how many different seating arrangements are there if everyone shows up?

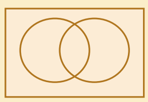
How many different predicates are there on $\Sigma = \{a,\ldots,z\}$?

Important cases of the Product Rule

- Cartesian product
  – $|A_1 \times A_2 \times \ldots \times A_n| = |A_1||A_2|\ldots|A_n|$
- Subsets of a set S
  – $|\boldsymbol{P}(S)| = 2^{|S|}$
- Strings of length n over Σ
  – $|\Sigma^n| = |\Sigma|^n$

Inclusion-Exclusion Principle

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

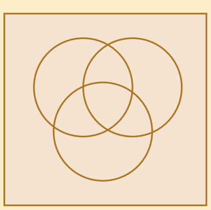


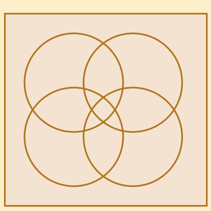
- How many binary strings of length 9 start with 00 or end with 11

Inclusion-Exclusion

- A class has of 40 students has 20 CS majors, 15 Math majors. 5 of these students are dual majors. How many students in the class are neither math, nor CS majors?

Generalizing Inclusion Exclusion



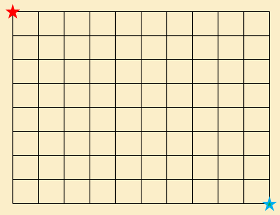


Permutations vs. Combinations

- How many ways are there of selecting 1^{st}, 2^{nd}, and 3^{rd} place from a group of 10 sprinters?
- How many ways are there of selecting the top three finishers from a group of 10 sprinters?

Counting paths

- How many paths are there of length n+m-2 from the upper left corner to the lower right corner of an n × m grid?



Binomial Coefficient Identities from the Binomial Theorem

$$\sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k = (x+y)^n$$

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0$$

$$\sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n$$

Combinations with repetition

- How many different ways are there of selecting 5 letters from {A, B, C} with repetition

How many non-decreasing sequences of {1,2,3} of length 5 are there?

How many different ways are there of adding 3 non-negative integers together to get 5 ?

1 + 2 + 2	• | • • | • •
2 + 0 + 3	• • | | • • •
0 + 1 + 4	
3 + 1 + 1	
5 + 0 + 0	

Probability

- Viewed as a very important topic for some subareas of Computer Science
  - Students required to take a statistics course
  - Some faculty want to add Probability for Computer Scientists
- Students will have seen the topics many times previously
- Discrete probability is a direct application of counting
- Advanced topics included (Bayes' theorem)

Goals

- Provide a domain for practicing counting techniques
- Remind students of a few probability concepts
  - Sample space, event, distribution, independence, conditional probability, random variable, expectation
- Introduce an advanced topic to see what is to come in other classes
- Understand applications of linearity of expectation

Discrete Probability

Experiment: Procedure that yields an outcome

Sample space: Set of all possible outcomes

Event: subset of the sample space

S a sample space of equally likely outcomes,
E an event, the probability of E, p(E) = |E|/|S|

Example: Poker

Probability of 4 of a kind

Discrete Probability Theory

- Set S
- Probability distribution p : S $\rightarrow$ [0,1]
  - For s $\in$ S, $0 \leq p(s) \leq 1$
  - $\Sigma_{s\in S}\, p(s) = 1$
- Event E, $E \subseteq S$
- $p(E) = \Sigma_{s\in E}\, p(s)$

Conditional Probability

Let E and F be events with p(F) > 0. The conditional probability of E given F, defined by p(E | F), is defined as:

$$p(E \mid F) = \frac{p(E \cap F)}{p(F)}$$

Random Variables

A random variable is a function from a sample space to the real numbers

Bayes' Theorem

Suppose that E and F are events from a sample space S such that p(E) > 0 and p(F) > 0. Then

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})}$$

False Positives, False Negatives

Let D be the event that a person has the disease

Let Y be the event that a person tests positive for the disease

$$p(Y \mid D) \quad p(D \mid Y)$$

$$p(\overline{Y} \mid D) \quad p(D \mid \overline{Y})$$

$$p(\overline{Y} \mid \overline{D}) \quad p(\overline{D} \mid \overline{Y})$$

$$p(Y \mid \overline{D}) \quad p(\overline{D} \mid Y)$$

Testing for disease

Disease is very rare: p(D) = 1/100,000

Testing is accurate:
False negative: 1%
False positive: 0.5%

Suppose you get a positive result, what do you conclude?

$$p(D \mid Y) = \frac{p(Y \mid D)p(D)}{p(Y \mid D)p(D) + p(Y \mid \overline{D})p(\overline{D})}$$

$$p(D) = 0.00001$$

$$p(Y \mid D) = 0.99$$

$$p(\overline{Y} \mid \overline{D}) = 0.995$$

P(D | Y) is about 0.002

Spam Filtering

From: Zambia Nation Farmers Union [znfukabwe@mail.zamtel.zm]
Subject: Letter of assistance for school installation
To: Richard Anderson

Dear Richard,
I hope you are fine, Iam through talking to local headmen about the possible assistance of school installation. the idea is and will be welcome.
I trust that you will do your best as i await for more from you.
Once again
Thanking you very much
Sebastian Mazuba.

Expectation

The expected value of random variable X(s) on sample space S is:

$$E(X) = \sum_{s \in S} p(s)X(s)$$

$$E(X) = \sum_{r \in X(S)} p(X = r)r$$

Left to right maxima

```
max_so_far := A[0];
for i := 1 to n-1
     if (A[ i ] > max_so_far)
         max_so_far := A[ i ];
```

5, 2, 9, 14, 11, 18, 7, 16, 1, 20, 3, 19, 10, 15, 4, 6, 17, 12, 8

Relations

- Some of this material is highly relevant
  - Relational database theory
  - Difficult to cover the material in any depth
- Large number of definitions
  - Easy to generate homework and exam questions on definitions
  - Definitions without applications unsatisfying

Goals

- Convey basic concepts of relations
  - Sets of pairs
  - Relational operations as set operations
- Understand composition of relations
- Connect with real world applications

Definition of Relations

Let A and B be sets,
A binary relation from A to B is a subset of $A \times B$

Let A be a set,
A binary relation on A is a subset of $A \times A$

Combining Relations

Let R be a relation from A to B
Let S be a relation from B to C
The composite of R and S, $S \circ R$ is the relation from A to C defined

$S \circ R = \{(a, c) \mid \exists$ b such that $(a,b) \in R$ and $(b,c) \in S\}$

Powers of a Relation

$R^2 = R \circ R = \{(a, c) \mid \exists$ b such that $(a,b) \in R$ and $(b,c) \in R\}$

$R^0 = \{(a,a) \mid a \in A\}$

$R^1 = R$
$R^{n+1} = R^n \circ R$

How is  related to  ?

From the Mathematics Geneology Project

Erhard Weigel
Gottfried Leibniz
Jacob Bernoulli
Johann Bernoulli
Leonhard Euler
Joseph Lagrange
Jean-Baptiste Fourier
Gustav Dirichlet
Rudolf Lipschitz

Felix Klein
C. L. Ferdinand Lindemann
Herman Minkowski
Constantin Caratheodory
Georg Aumann
Friedrich Bauer
Manfred Paul
Ernst Mayr
Richard Anderson

http://genealogy.math.ndsu.nodak.edu/

n-ary relations

Let $A_1, A_2, \ldots, A_n$ be sets. An n-ary relation on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$.

Relational databases

Student_Name	ID_Number	Major	GPA
Knuth	328012098	CS	4.00
Von Neuman	481080220	CS	3.78
Von Neuman	481080220	Mathematics	3.78
Russell	238082388	Philosophy	3.85
Einstein	238001920	Physics	2.11
Newton	1727017	Mathematics	3.61
Karp	348882811	CS	3.98
Newton	1727017	Physics	3.61
Bernoulli	2921938	Mathematics	3.21
Bernoulli	2921939	Mathematics	3.54

Alternate Approach

Student_ID	Name	GPA
328012098	Knuth	4.00
481080220	Von Neuman	3.78
238082388	Russell	3.85
238001920	Einstein	2.11
1727017	Newton	3.61
348882811	Karp	3.98
2921938	Bernoulli	3.21
2921939	Bernoulli	3.54

Student_ID	Major
328012098	CS
481080220	CS
481080220	Mathematics
238082388	Philosophy
238001920	Physics
1727017	Mathematics
348882811	CS
1727017	Physics
2921938	Mathematics
2921939	Mathematics

Database Operations

Projection

Join

Select

Matrix representation

Relation R from $A=\{a_1, \ldots a_p\}$ to $B=\{b_1, \ldots b_q\}$

$$m_{ij} = \begin{cases} 1 \text{ if } (a_i, b_j) \in R, \\ 0 \text{ if } (a_i, b_j) \notin R. \end{cases}$$

{(1, 1), (1, 2), (1, 4), (2,1), (2,3), (3,2), (3, 3) }

Graph Theory

- End of term material – limited chance for homework
- Cannot ask deep questions on the exam
- Graph theory is split across three classes
  - Algorithmic material is covered in other classes

Goals

- Understand the basic concept of a graph and associated terminology
- Model real world with graphs
  - Real world to formalism
- Elementary mathematical reasoning about graphs

Graph Theory

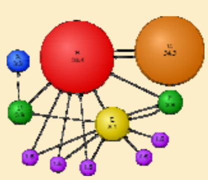
- Graph formalism
  - G = (V, E)
  - Vertices
  - Edges
- Directed Graph
  - Edges ordered pairs
- Undirected Graph
  - Edges sets of size two

Big Graphs

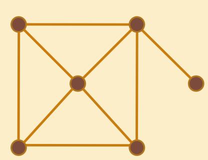
- Web Graph
  - Hyperlinks and pages
- Social Networks
  - Community Graph
    - Linked In, Face Book
  - Transactions
    - Ebay
  - Authorship
    - Erdos Number

Page Rank

- Determine the value of a page based on link analysis
- Model of randomly traversing a graph
  - Weighting factors on nodes
  - Damping (random transitions)



Degree sequence



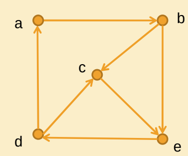
- Find a graph with degree sequence
  - 3, 3, 2, 1, 1
- Find a graph with degree sequence
  - 3, 3, 3, 3, 3

Handshake Theorem

$$2e = \sum_{v \in V} \deg(v)$$

Counting Paths

Let A be the Adjacency Matrix. What is A^2?



$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$