CS 212

Mathematical Foundations of Computer Science

Finishing Public-Key Cryptography and Final Review

The Revolution: Public Key Cryptography

- Private Key Crypto is impractical. Key should be as large as message (information theory) !

Public key

Can be used to lock, but not unlock

Private key

- What if the key for locking is public?
- Anyone can encrypt & send messages to Receiver !
- But only Receiver can decrypt!
  (only Receiver has the key to open the lock)

Encryption & Decryption

$n = pq$. e chosen s.t. $\gcd\big(e, (p-1)(q-1)\big) = 1$.

$\boxed{\text{Public key: } (e, n)}$

d chosen s.t. $de \equiv 1 \; mod \; \big((p-1)(q-1)\big)$.

Private key: $\big(d, \phi(n) = (p-1)(q-1)\big)$

Message=m

Encryption (Sender S):

1. Pick message $m \in \{1, \ldots, n-1\}$. Verify that $\gcd(m, n) = 1$.

2. Compute $c = m^e \; (mod \; n)$ and send c.

Decryption (Receiver R):

R has private key $(d, \phi(n))$

Find $c^d \; (mod \; n)$. Claim: $m = c^d (mod \; n)$.

How is this implemented?

{primes that are ≤ m} $\sim \frac{m}{\log m}$

1. Generate primes (say 512 bits) p, q .

2. Generating e, how do we find $d = \mathrm{e}^{-1} (\mathrm{mod}\ \phi(n))$?

Exists because $\gcd(e, \phi(n)) = \underline{1}$.

Inverse can be computed using Euclid's algorithm.

3. How do we compute m^e? Say m^{31}?

4. $\mathrm{m}^{\mathrm{e}} (\mathrm{m}od\ n)$?

Repeated Powering

3. How do we compute m^e? Say m^{32}? m^{31}?

$m^{32} = m * m * m \underline{\qquad\qquad\qquad} * m$

$m^{32} = (m^{16})^2 = ((m^8)^2)^2 = ((((m^4)^2)^2)^2$

$\rightarrow$ square 5 times

m^{31}

$(31)_{10} = (1\,0\,1\,1\,1)_2$ $m^{31} = m^{16} * 1 * m^4 * m^2 * m$

4. $\mathrm{m^e}(\mathrm{m}od\ n)$?

Why is this secure?

- Evesdropper gets access to public key, ciphertext: $n, e, m^e (mod\ n)$.
- Can we compute m?
- If we can factor n, we can compute p, q and hence d

  $\phi(n)$

  i.e. $e^{-1}\ mod\big((p-1)(q-1)\big)$

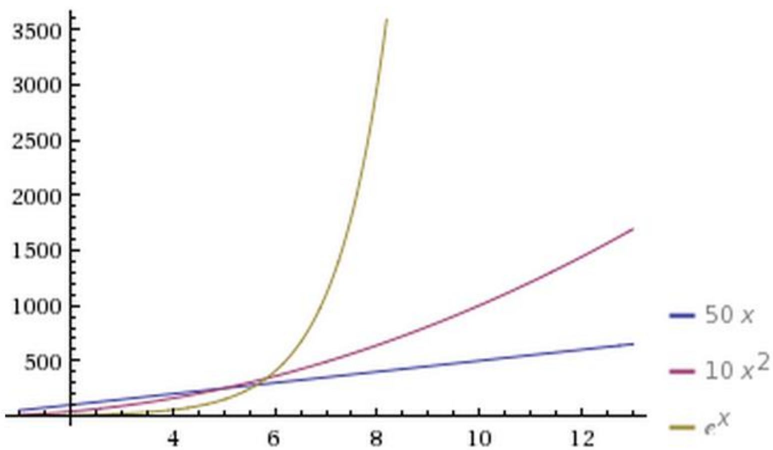
  $d = e^{-1} (mod\ \phi(n))$

Thought to be as hard as factoring!

How hard is factoring? Any algorithm that runs in time polynomial in number of bits?

Polynomial Time Algorithms

It ain't good if ain't snappy enough



$O(n)$?
$O(n^2)$?
$O(n^{10})$?

polynomial time $O(n^c)$ for some constant c

$O(n^{\log n})$?
$O(2^n)$?
$O(n!)$?

non-polynomial time

Polynomial time algorithm is efficient

Exponential time algorithm is inefficient

Easy vs Hard Problems

Problems with efficient algorithms:

- Graph Matching, 2-Coloring a graph
- Maximum Flow, Minimum Cut.
- Testing Primality

Problems with no known efficient algorithms:

- Integer Factoring
- 3-Coloring a graph (assignment problem in PS6)

Checklists

Announcements

1. Final Exam on Monday, Dec ~~6~~ 5th at 3pm – 4:50pm.
2. You will be allowed one regular sheet (A4/ letter) of paper to write down formulae etc.
3. Portions: everything covered up and until Monday, Nov 28th.
4. Any student with particular needs (like extended time) should contact instructors and sign up on ANU.

Number Theory

GCD and its properties

$k|n$ means n is a multiple of k.

integers

Even numbers: $2k$; Odd numbers: $2k-1$ for $k \in \mathbb{Z}$

$\gcd(a,b)$: largest common divisor between a and b

- $\gcd(a,b)$ is smallest +ve integer combination of a, b (Thm)
- Every common divisor of a, b divides $\gcd(a,b)$
- $\gcd(ka, kb) = k \times \gcd(a,b)$

Relatively prime: $\gcd(a,b)=1$

- $\gcd(a,b) = 1, \gcd(a,c) = 1 \Rightarrow \gcd(a,bc) = 1$

Suppose b is prime
$\gcd(a,b)=1$ unless
a is a multiple of b.

- $\boxed{\gcd(a,b) = \gcd(b, remainder(a,b))}$

Modulo arithmetic

$a\ (mod\ n) \equiv r \iff a = dn + r$ for some integer d

Congruence modulo n: $a \equiv b\ (mod\ n)$ iff $n \mid (a - b)$

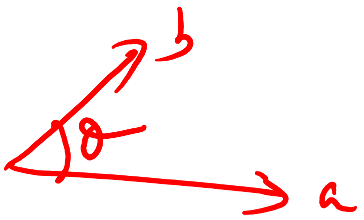
- Addition:
  $$a \equiv b\ (mod\ n),\ c \equiv d\ (mod\ n) \Rightarrow a + c \equiv b + d\ (mod\ n)$$
- Multiplication:
  $$a \equiv b\ (mod\ n),\ c \equiv d\ (mod\ n) \Rightarrow ac \equiv bd\ (mod\ n)$$
- Residue classes (mod n):

  0 mod 3 ≡ { …, -6, -3, 0, 3, 6, ..} mod 3
  1 mod 3 ≡ { …, -5, -2, 1, 4, 7, ..} mod 3
  2 mod 3 ≡ { …, -4, -1, 2, 5, 8, ..} mod 3

Linear Algebra

Vectors, Inner products & Matrices

- Vectors over real numbers in n dimensions i.e. n coordinates
- Inner product between vectors $\langle a, b \rangle = \|a\|_2 \|b\|_2 \cos(\theta)$
- Orthogonal vectors $\Leftrightarrow$ $\langle a, b \rangle = 0$



- Matrix multiplication
  (Matrix-matrix multiplication
  or Matrix-vector ")

Eigenvalues, Eigenvectors

Given any matrix $M \in \mathbb{R}^{n\times n}$, $e \in \mathbb{R}^n$ is an eigenvector iff for some scalar $\lambda \in \mathbb{R}$, $\boldsymbol{Me = \lambda\, e}$.

$(\lambda, \boldsymbol{e})$ is called an eigenvalue, eigenvector pair.

$(\lambda, 2e)$

$M(2e) = 2Me = 2\lambda e$
$= \lambda(2e)$

Simple Fact:

$M \in \mathbb{R}^{n\times n}$
$x \in \mathbb{R}^n$

$$\underbrace{x^T M x}_{1\times 1} = \sum_{i=1,j=1}^{n} M_{ij}\, x_i x_j$$

$(x^T)_{1\times n}\, M_{n\times n}\, (x)_{n\times 1}$

$(\;)_{1\times 1}$

Eigendecompositions

Roughly: Eigenvalues and eigenvectors are the building blocks that make up matrices

Spectral Thm. For any $n \times n$ symmetric matrix M (over reals)

1. All its eigenvalues are real.
2. Further, there are n real eigenvalues (and eigenvectors)

$$(\lambda_1, e_1), (\lambda_2, e_2) \dots (\lambda_n, e_n),$$

such that every pair of eigenvectors is orthogonal i.e.

$$\langle e_i, e_j \rangle = 0 \text{ for } i \neq j.$$

$\|e_i\|_2^2 = \langle e_i, e_i \rangle = 1$

3. In addition,

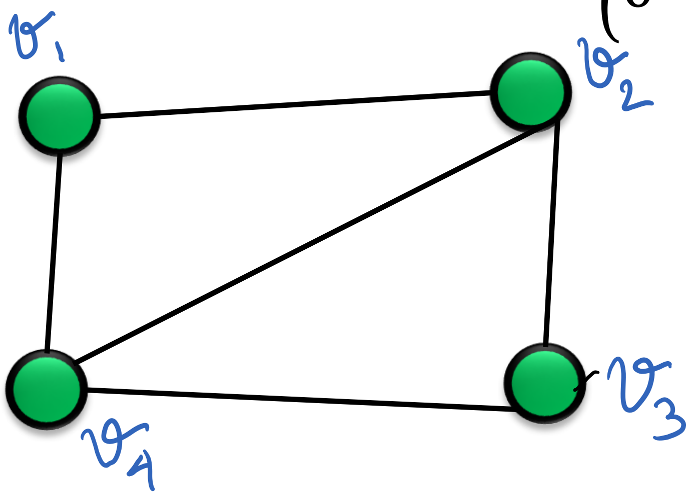
$$\boxed{M = \sum_{i=1}^{n} \lambda_i e_i e_i^T}$$

Adjacency Matrices

Graph G with n vertices.

The adjacency matrix is the n x n matrix A=[a_{ij}] with:

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge} \\ 0 & \text{if } (v_i, v_j) \text{ is not an edge} \end{cases}$$



Undirected graphs: Symmetric matrix. Hence, all eigenvalues exist, and eigenvectors for diff. eigenvalues are orthogonal to each other.

Fact: The number of walks of length k from node i to node j is the entry in position (i, j) in the matrix A^k

Linear Programming

Linear Programs

Variables: $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$

Constraints: given by $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

$$\begin{aligned} &\max \quad c^T x = \sum_{i=1}^{n} c_i x_i \\ \text{such that} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

$$\underset{m}{\left[\, \overset{n}{A} \,\right]} \underbrace{\left[\, x \,\right]}_{n \times 1} \leq \underbrace{\left(\, b \,\right)}_{m \times 1}$$

Example:

Maximize $5x_1 - 3x_2$

$$x_1 + 3x_2 \leq 5$$
$$3x_1 + x_2 \leq 4$$
$$4x_1 - 8x_2 \leq -4$$
$$x_1, x_2 \geq 0$$

LP Formulation

E.g.: $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $b = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Variables: $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$

Constraints: given by $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

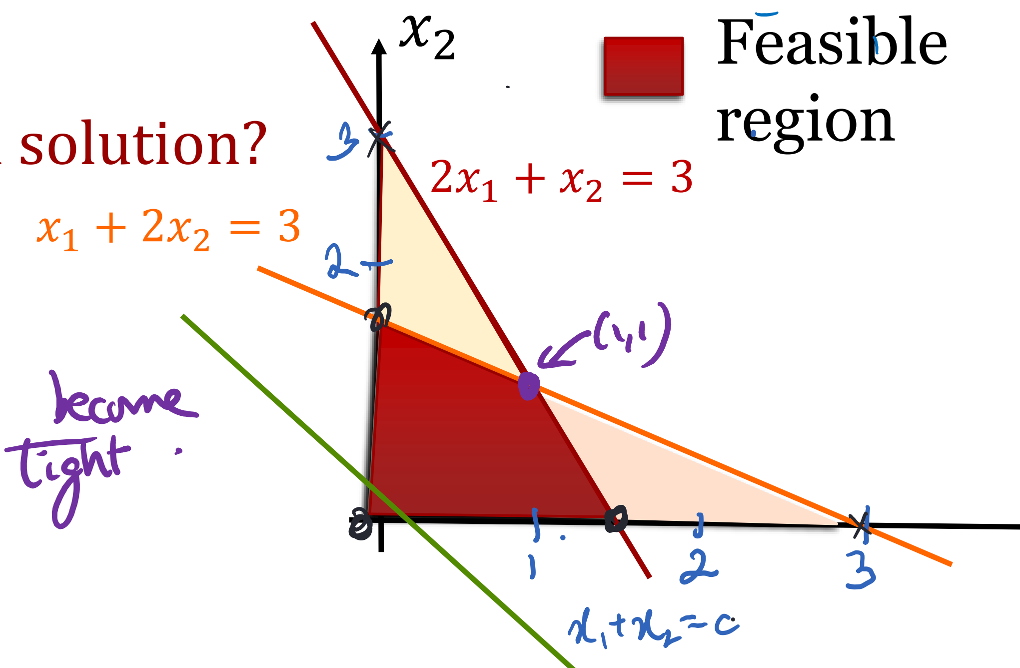
Standard ~~General~~ form

$$\max \quad c^T x$$

$$\text{s.t.} \qquad Ax \le b, x \ge 0$$

What can we say about optimal solution?

One of the corner points.



What is a corner point?

pt where n of the inequalities become tight.

"Standard" LP Formulation

Variables: $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$

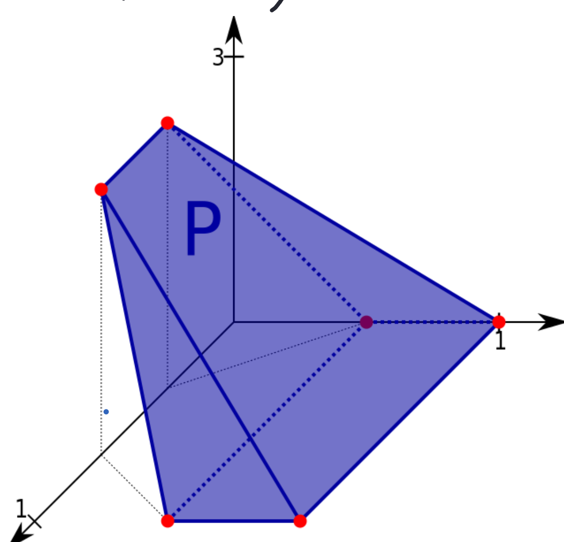
Constraints: given by $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ & given $c \in \mathbb{R}^n$

$$\max \quad c^T x = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

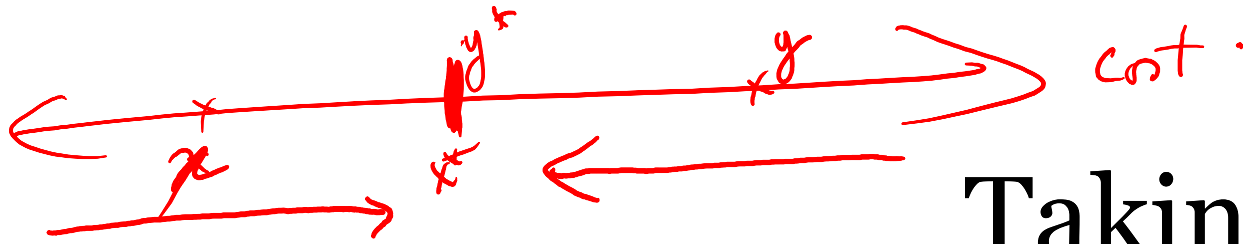
$$\text{such that} \quad Ax \le b \qquad \forall i \in [m]: \sum_{j=1}^{n} a_{ij} x_j \le b_j$$

$$x \ge 0 \qquad \forall j \in [n]: x_j \ge 0$$

$\le$



Claim: Standard LP formulation can capture general Linear programs.



Taking Duals

- This Procedure possible for any LP!

Primal LP:

$$\max \quad \sum_{j=1}^{n} c_j x_j$$

s.t $\forall i \in [m]$

$$\sum_{j=1}^{n} a_{ij}\, x_j \leq b_i \quad \} y_i$$

$$x \geq 0$$

Dual LP:

$$\min \quad \sum_{i=1}^{m} b_i y_j$$

s.t $\forall j \in [n]$

$$\sum_{i=1}^{n} a_{ij}\, y_i \geq c_j$$

$$y \geq 0$$

- One dual variable for each primal constraint.
- One dual constraint for each primal variable.

Primal LP: $\max \quad c^T x$

such that $\quad Ax \leq b$

$\quad x \geq 0$

= Strong duality

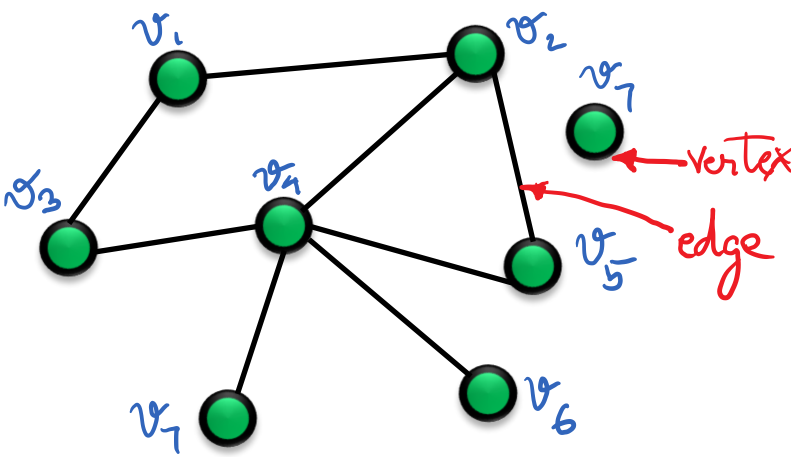
Dual LP: $\min \quad b^T y$

such that $\quad A^T y \geq c$

$\quad y \geq 0$

Graph Theory

Graphs



$G(V, E)$

Graph $G = (V, E)$ is a pair of sets

$V = \mathit{set\ of\ vertices},$

$\mathrm{E} \subset V \times V = \text{set of edges}$

Undirected graph: For any $u, v \in V$, if $(u, v) \in E$, then $(v, u) \in E$

$|V| = n$ #nodes in a graph

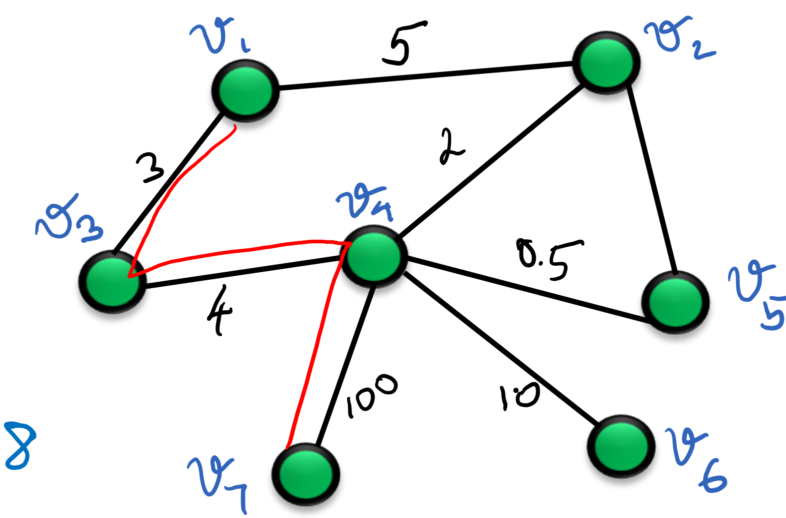
$|E| = m$ #edges in a graph

- degree of a vertex, regular graphs. Fact: $\sum_{v \in V} d_v = 2\,|E|$

G(V,E) is a simple graph iff it is undirected, with no self-loops and no parallel edges.

Weighted Graphs, Distances

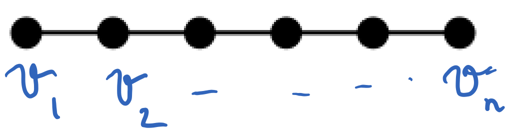
Weighted graphs: $G(V, E, w)$



$G(V, E)$

Edges have numbers associated with them, representing extent of relation e.g. maps with distances.

Path:



$|E| = 8$

- Weights can also encode distances.
- Length of path = sum of weights of edges on the path

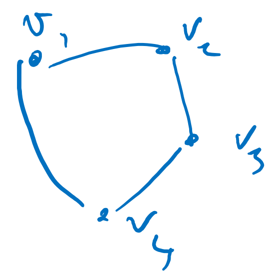
lngth $v_1 - v_3 - v_4 - v_7 = 107$.

- Distance between u,v:

$d(u, v)$ = length of shortest path u *to* v

Connectivity, Trees

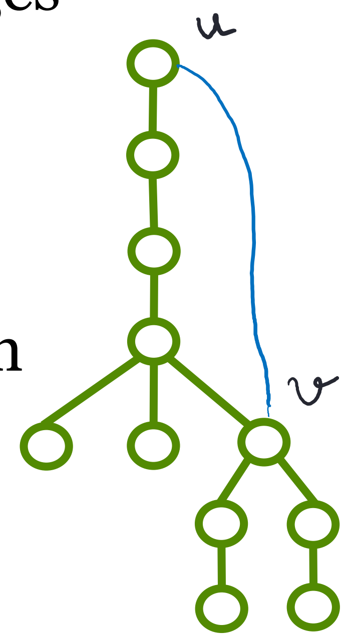
- Connectivity, Connected graphs
- Connected components
- Paths, Cycles.
- Trees: Connected graphs with no Cycles.
- Number of edges in a tree = n-1.
- Spanning Trees, Minimum Spanning Trees.



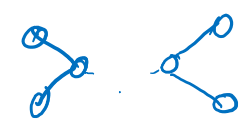
Equivalent Definitions of Trees

Theorem: Let G be a graph with n vertices and m edges

The following are equivalent:



1. G is a connected and acyclic (i.e. G is a tree)
2. Every two vertices of G are joined by a unique path
3. G is connected and $m = n - 1$
4. G is acyclic and $m = n - 1$
5. G is acyclic and if any two non-adjacent nodes are joined by an edge, the resulting graph has exactly one cycle



Bipartite Graphs

- Graph Coloring

A graph is bipartite iff it is 2-colorable i.e. the nodes can be partitioned into two sets V_L and V_R such that *all edges* go only between V_L and V_R i.e. no edge inside V_L or inside V_R

Theorem: A graph $G(V, E)$ is bipartite iff there is G has no odd cycle.

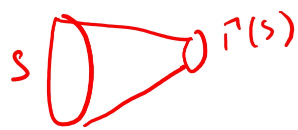
Matchings

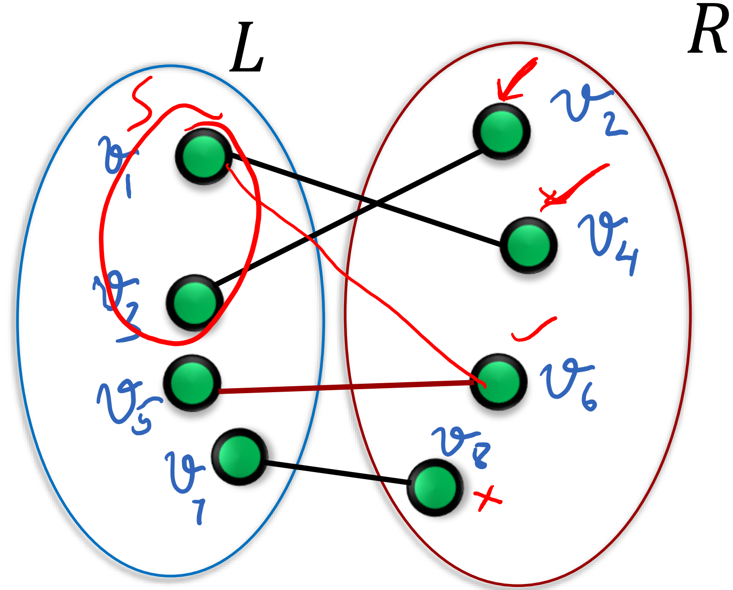
$|N(S)| \geq |S|$

Matching: A set of edges, no two of which share a vertex.

Matching saturating L:
includes every vertex in L

Perfect Matching: A matching is perfect if it includes every vertex in L and R.





Theorem. Bipartite graph $G(V = (L, R), E)$ has *a matching saturating L* iff for any subset $S \subseteq L$, there are at least $|S|$ nodes of R connected to at least one node in S. for perfect matching, additionally $|L| = |R|$

Planar Graphs

A graph is planar if it can be drawn (represented) on the plane without any crossing edges (no edges intersect).

Faces

Thm. If G is a connected planar graph G with vertex set V (size n), edges E (m of them) and faces F (f of them), then

$$|\mathrm{V}| - |\mathrm{E}| + |F| = n - m + f \;=\; 2$$

connected

Thm. In a planar graph $G(V, E)$ on $n \geq 3$ vertices, the number of edges $|E| \leq 3n - 6$.

Relation between graph properties

- What are Cliques?
- What are Independent Sets?
- Graph Complement $\bar{G}$?

Fact: Independent sets in G are Cliques in $\bar{G}$ and vice-versa.

Growth Rate of Functions

Big-Oh Notation

- $g(n) = \Omega(f(n))$: g is asymptotically lower bounded by f(n)

$$g(n) = \Omega\big(f(n)\big) \text{ iff } f(n) = O(g(n)).$$

- $g(n) = \Theta(f(n))$: f and g are asymptotically of the same order of magnitude (same upto constant factors)

$$g(n) = \Theta\big(f(n)\big) \text{ iff } f(n) = O(g(n)) \text{ and } g(n) = O(f(n)).$$

If $\lim_{n\to\infty} \frac{f(n)}{g(n)} \le c < \infty$ for some constant c ≥ 0, then $f(n) = O(g(n))$

If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, then $f(n) = o(g(n))$

Simple Rules for Big-Oh

1. Constant factors don't matter: $c \cdot f(n) = \Theta(f(n))$

2. Smaller terms don't matter e.g. $an^2 + bn + c = \Theta(n^2)$.

3. Among polynomials, exponent is most important.

   i.e. $n^a = o(n^b)$ if $a < b$.

4. Logarithms are dominated by polynomials (use L'Hopital rule)

   i.e. $\log n \quad = o(n^a)$ if $a > 0$

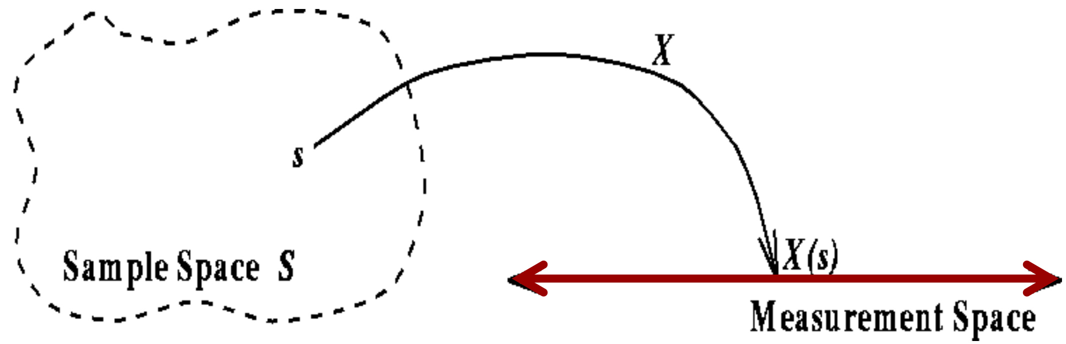
5. If $f(n) = o\big(g(n)\big)$, then $2^{f(n)} = o(2^{g(n)})$

Fact: Use $n^a = 2^{a \log n}$ to put polynomials in exponent form

Probability Recap/ Checklist

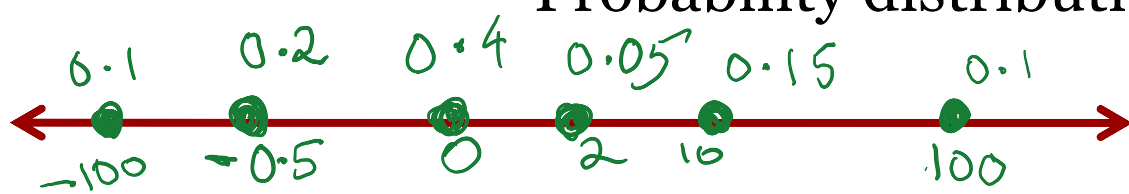
Recap: Random Variables

1. Think of a R.V. as function from S to the reals ℝ (input to the function is random)



2. Or think of the induced distribution on ℝ, randomness is "pushed" to the values of the function.

Probability distribution on ℝ



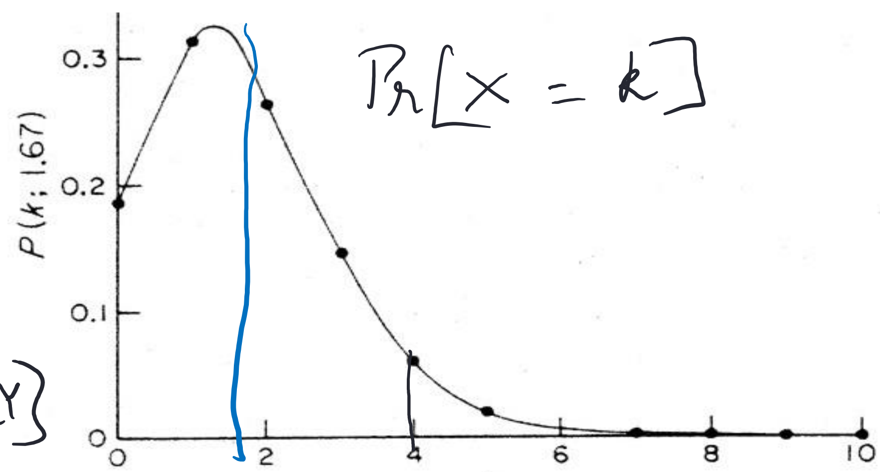
Measurement space ℝ

Expectation

Average/mean value of the random variable X

The expectation, or expected value of a random variable X is

$$E[X] = \sum_{t \in S} Pr(t) \times X(t) = \sum_{k} k \times \Pr[X = k]$$



If X and Y are random variables ,

$$E[aX + bY] = aE[X] + bE[Y]$$

If X & Y independent

$$Var[X+Y] = Var[X] + Var[Y]$$

Conditional Probability

The probability of event A given event B is written as $\Pr[A \mid B]$ `

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]} = \frac{\Pr[B|A]\Pr[A]}{\Pr[B]}$$

Conditional expectation of r.v. X given event B:

$$E[X \mid B] = \sum_{k} \Pr[X = k \mid B] \times k$$

Independent Events

A and B are independent events iff

1. $\Pr[\,A \mid B\,] = \Pr[\,A\,] \Leftrightarrow \Pr[\,A \cap B\,] = \Pr[\,A\,]\Pr[\,B\,]$

RVs X and Y are independent iff for every a,b, the events X=a and Y=b are independent

2. $E[X \cdot Y] = E[X] \cdot E[Y]$

3. $Var[X + Y] = Var[X] + Var[Y]$

Markov's inequality

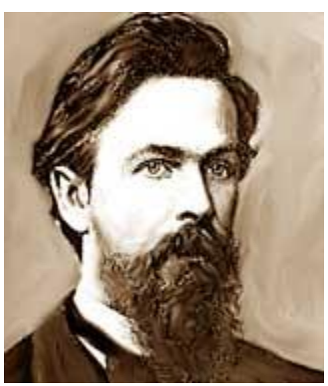
If X is a *non-negative r.v.* with mean E[X], then

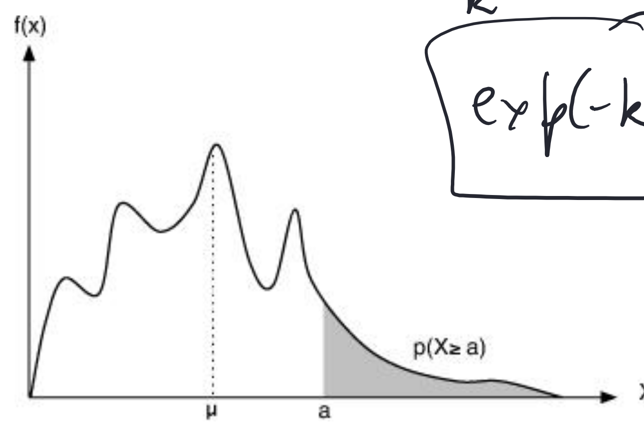
$$\Pr[\, X > 2E[X] \,] \leq ½$$

$$\Pr[\, X > k \cdot E[X] \,] \leq 1/k$$

$\frac{1}{k^2} = o\left(\frac{1}{k}\right)$

$\boxed{\exp(-k)} = o\left(\frac{1}{k}\right)$





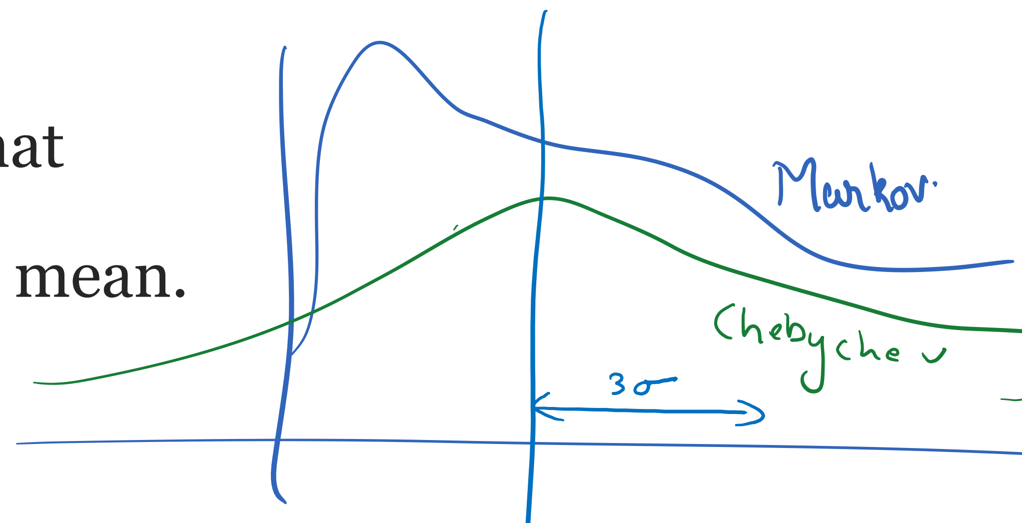
Chebychev Inequality

Bounds deviation around mean on both sides: X *any* random variable with mean $E[X]$, standard deviation $\sigma = \sqrt{Var[X]}$.

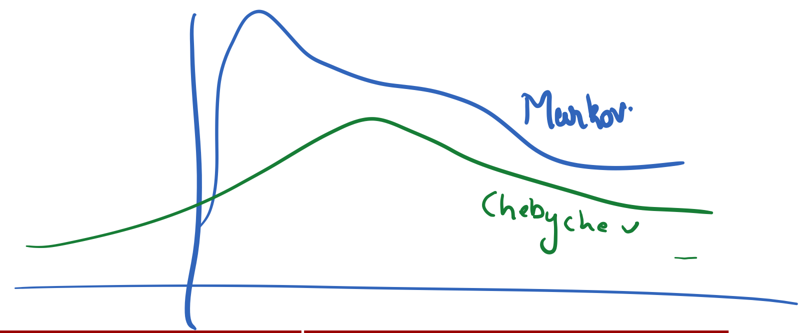
$$\Pr[|X - E[X]| > t\sigma] \leq 1/t^2$$

Eg. At most probability 0.1 that

X is more than 3σ away from mean.



A Comparison



Aspect	Markov Inequality	Chebychev Inequality	Chernoff Bounds
Pre-conditions	non-neg r.v	any r.v	sums of independent indicator r.v.

Recap of Basic Set Theory and Relations to Probability

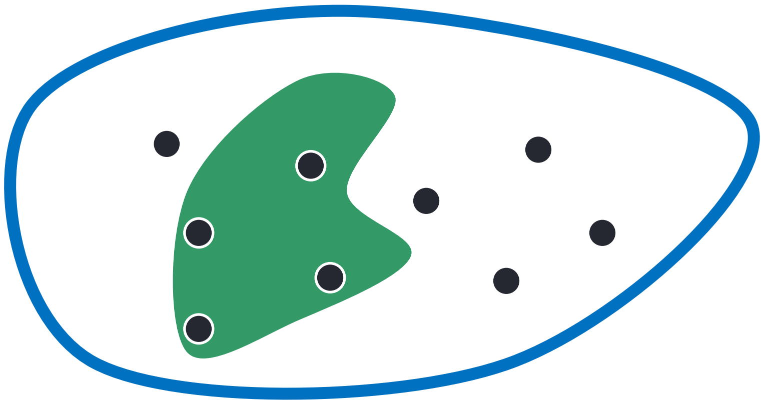
Elements, Universe

A **set** S is an unordered collection of objects, where each *element* of the set is considered to included only once.

Set Builder Notation:

$\{x^2 \mid x \in \mathbb{Z}\}$

$\{x \in \mathbb{R} \mid \cos x > 0.75\}$

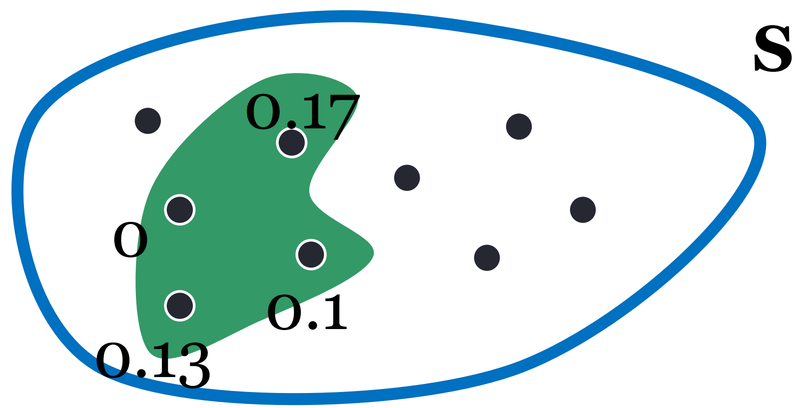


Subset $T \subset S$ is an subcollection of these elements.

Universe U is set of all possible elements

Probability Equivalents

Sample Space: a (finite) probability distribution D is a finite set S of elements, where each element t in S has a non-negative real weight or probability $p(t)$

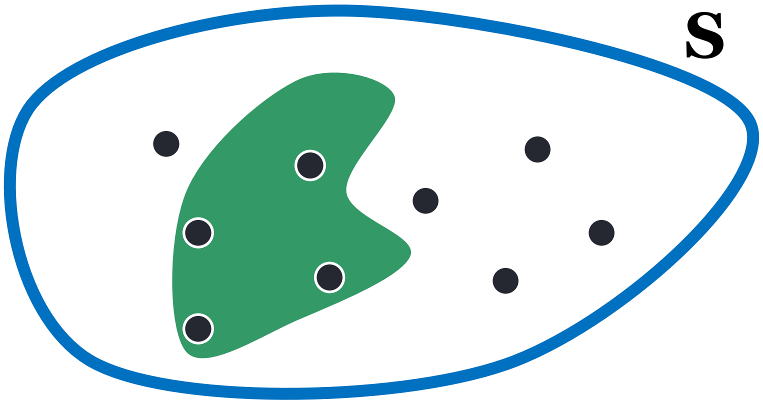


- Any set E ⊆ S is called an Event.
- Elements called Atomic events.

$\Pr_D[E] = 0.4$

$$\Pr_D[E] = \sum_{t \in E} p(t)$$

Basic Set Operations



Universe is S.

T^C: complement of T.

Everything in S that does not belong to T.

Counting:

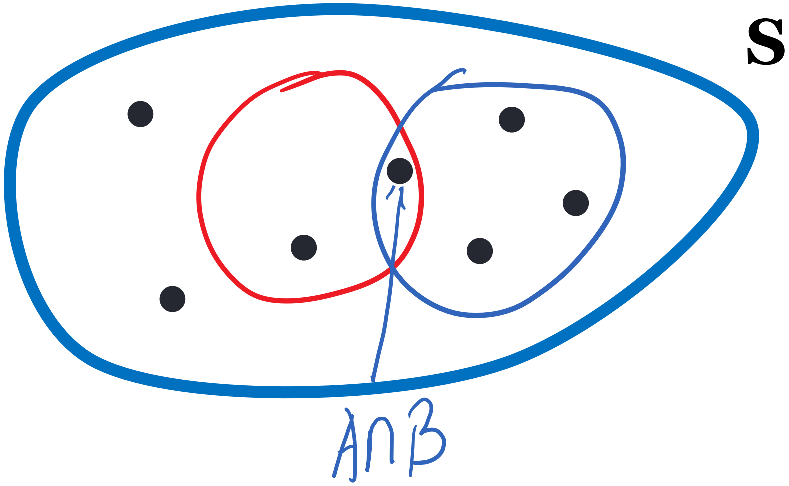
$$|\bar{T}| = |S| - |T|$$

Probability: event A

$$\text{Pr[A not occuring]} = \Pr[\bar{A}] = 1 - \Pr[A]$$

Intersection (AND)

Set Theory: Elements in both A and B



Probability:

$$\Pr[\, A \cap B \,] = \Pr[\, A|B \,]\, \Pr[B]$$

A and B are independent events iff

1. $\Pr[\, A \mid B \,] = \Pr[\, A \,] \quad \Leftrightarrow \quad \Pr[\, A \cap B \,] = \Pr[\, A \,]\, \Pr[\, B \,]$

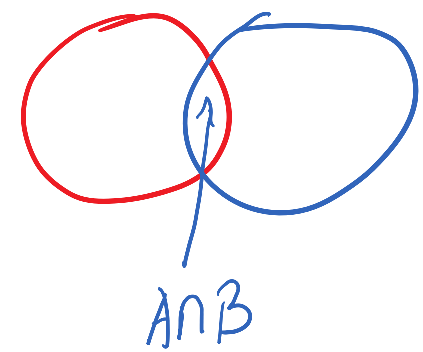
2. $E[X \cdot Y] = E[X] \cdot E[Y]$

Union (OR)

Sum Rule: If A and B are disjoint events, then $|A \cup B| = |A| + |B|$

If A and B are sets, then

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



Probability: If A and B are events, then

$$\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$$

More events (k): Inclusion-exclusion formula in general.

Union Bound (Boole's inequality): For events $A_1, A_2, \dots, A_k$

$$\Pr\left[\cup_{i=1}^{k} A_i\right] \le \Pr[A_1] + \Pr[A_2] + \cdots + \Pr[A_k]$$

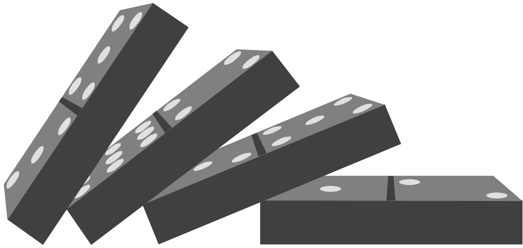
Questions?

Induction Recap

Recap of Proofs

- **Proof by Contradiction**: assume the negation of the statement and derive contradiction.

- **Principle of Mathematical Induction**



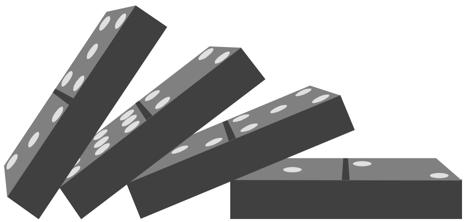
To prove: For all $k \in \mathbb{N}$, predicate $P(k)$ is true.

1. Base case: *Establish that $P(0)$ is true.*
2. For all $k \in \mathbb{N}$: $P(k) \implies P(k+1)$

Assume that P(k) is true. Establish that P(k+1) is true.

Strong Induction

To prove: For all $\mathrm{k} \in \mathbb{N}$, predicate $P(k)$ is true.



Steps:

1. Base case: *Establish that* $P(0)$ *is true.*
2. Assume that $P(1), \ldots, P(k-1)$ is true (Inductive Hypothesis).
3. Derive that $P(k)$ is true.

$$\text{i.e.} \quad P(1), \ldots, P(k-1) \Rightarrow P(k)$$

By Strong Induction, $P(k)$ is true for all $k \in \mathbb{N}$.

Picking Induction Statements, Invariants

1. Often, to prove a statement inductively you may have to prove a stronger statement first!
2. Work out examples for small values of k

Invariants:

- Not varying; constant. Unaffected by any operation.
- P(k): “Invariant holds in kth step of the algorithm”
- Show using Mathematical Induction

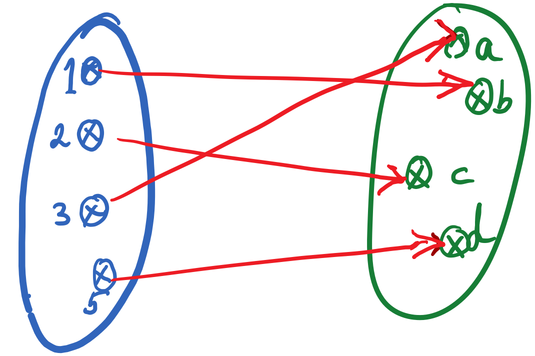
Example: Insertion Sort Algorithm for sorting n numbers. After first k steps, the first k numbers are sorted.

Questions?

Counting Recap

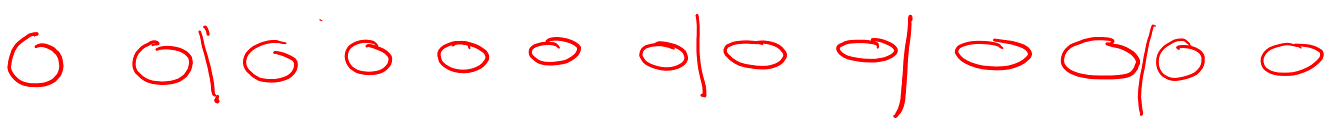
Counting using Bijections

Bijective function: one-to-one onto mapping



Thm. If $f : A \rightarrow B$ is a bijective function, then $|A|=|B|$

Five kinds of donuts. Number of many ways to select a dozen



= number of ways of splitting 12 identical objects into 5 groups
= number of 16 bit sequences with exactly 4 ones

Simple Rules : Sum Rule, Product rule

Sum Rule: If A and B are disjoint events, then $|A \cup B| = |A| + |B|$

Product Rule: If there are two unrelated events that have n_1 and n_2 possible outcomes respectively, the total number of possible outcomes is $n_1 n_2$.

Product Rule: For sets A, B, the crossproduct $|A \times B| = |A| \times |B|$

Given a set S of size n, how many different subsets of S? 2^n

Counting Formulae

1. Arranging n objects in k positions (without repetition): ${}^{n}P_{k} = \frac{n!}{(n-k)!}$

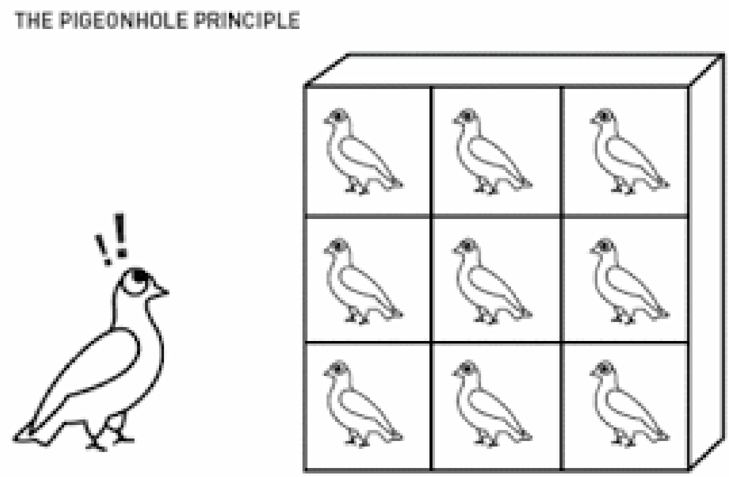
2. Filling k positions with n objects (with repetition): n^k

3. Selecting k out of n objects (no ordering, no repetition): $\binom{n}{k}$

4. Selecting n identical objects in k different bins: $\binom{n+k-1}{k-1}$

Pigeonhole Principle

If socks can be either red, blue or green, how many socks do we need to find a matching pair? Ans = 4



Pigeonhole Principle (basic):
If there are n pigeons and $\leq n - 1$ pigeon-holes, then there must be at least 2 pigeons in one of the holes.

Pigeonhole Principle (general):
If $|Y| = n$ and $|X| \geq nk + 1$, then for any function $f: X \rightarrow Y$, there exists a $y \in Y$ to which $k + 1$ elements from X map to.

Binomial Theorem

$$(a+b)^n = (a \overset{1}{+} b)(a \overset{2}{+} b)(a+b)\dots(a \overset{n}{+} b)$$

$$= 1 \cdot a^n + \binom{n}{1} a^{n-1}b + \cdots + \binom{n}{i} a^{n-i}b^i + \cdots + \binom{n}{n-1} ab^{n-1} + \binom{n}{n} b^n$$

$$\boxed{(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i}$$

Different forms:

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i$$

$a = 1$
$b = x$

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

$a = 1$
$b = -x$

Questions?

Good Luck!!