Counting Strategies: Inclusion-Exclusion, Categories

	Miles Jones	**MTThF 8:30-9:50am**	**CSE 4140**

August 19, 2016

Length n binary strings

Select which method lets us count the number of length n binary strings.

A. The product rule.
B. The sum rule.
C. Either rule works.
D. Neither rule works.

Select first bit, then second, then third …

{0…} U {1…} gives recurrence $N(n) = 2N(n-1)$, $N(0)=1$

$N(n) = 2^n$

Memory: storing length *n* binary strings

How many binary strings of length *n* are there? 2^n

How many bits does it take to store a length *n* binary string? n

Memory: storing length *n* binary strings

How many binary strings of length *n* are there? **2^n**

How many bits does it take to store a length *n* binary string? **n**

General principle: number of bits to store an object is

$$\lceil \log_2(\text{number of objects}) \rceil$$

Why the ceiling function?

Memory: storing integers

Scenario: We want to store a non-negative integer that has at most n digits.
How many bits of memory do we need to allocate?

A. n
B. 2^n
C. 10^n
D. $n*\log_2 10$
E. $n*\log_{10} 2$

Ice cream!

At an ice cream parlor, you can choose to have your ice cream in a bowl, cake cone, or sugar cone. There are 20 different flavors available.

How many single-scoop creations are possible?

A. 20
B. 23
C. 60
D. 120
E. None of the above.

Ice cream!

At an ice cream parlor, you can choose to have your ice cream in a bowl, cake cone, or sugar cone. There are 20 different flavors available.

You can convert your single-scoop of ice cream to a sundae. Sundaes come with your choice of caramel or hot-fudge. Whipped cream and a cherry are options. How many desserts are possible?

A. 20*3*2*2
B. 20*3*2*2*2
C. 20*3 + 20*3*2*2
D. 20*3 + 20*3*2*2*2
E. None of the above.

A scheduling problem

In one request, four jobs arrive to a server: J1, J2, J3, J4.

The server starts each job right away, splitting resources among all active ones.

Different jobs take different amounts of time to finish.

How many possible finishing orders are there?

4^4 $4 \cdot 4$ $4!$

A scheduling problem

In one request, four jobs arrive to a server: J1, J2, J3, J4.

The server starts each job right away, splitting resources among all active ones.

Different jobs take different amounts of time to finish. J2 ___ ___ ___ ___

How many possible finishing orders are there?

Which options are available will depend on first choice; but the **number** of options will be the same.

Product rule analysis

- 4 options for which job finishes first.
- Once pick that job, 3 options for which job finishes second.
- Once pick those two, 2 options for which job finishes third.
- Once pick first three jobs, only 1 remains.

$$\boxed{(4)(3)(2)(1) = 4! = 24}$$

Permutations

Permutation:

Rosen p. 407

rearrangement / ordering of n distinct objects so that each object appears exactly once

Theorem 1: The number of permutations of n objects is

$$\mathbf{n! = n(n-1)(n-2) \ldots (3)(2)(1)}$$

Convention: $0! = 1$

$1! = 1$

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

all 5! possible permutations

Must **start in New York** and **end in Seattle**.

How many ways can the trip be arranged?

A. 7!
B. 2^7
C. None of the above.

Traveling salesperson

4 !

Planning a trip to

New York
1 Chicago
Baltimore
~~Los Angeles~~
3 San Diego
4 Minneapolis
Seattle

Must **start in New York** and **end in Seattle**.
Must also **visit Los Angeles immediately after San Diego**.

How many ways can the trip be arranged now?

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

Treat LA & SD as a single stop.

(1)(4!)(1) = 24 arrangements.

Must **start in New York** and **end in Seattle**.
Must also **visit Los Angeles immediately after San Diego**.

How many ways can the trip be arranged now?

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

Must **start in New York** and **end in Seattle**.
Must also **visit Los Angeles and San Diego immediately after each other (in any order).**

How many ways can the trip be arranged now?

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

Break into two disjoint cases:
Case 1: LA before SD 24 arrangements
Case 2: SD before LA 24 arrangements

Must **start in New York** and **end in Seattle**.
Must also **visit Los Angeles and San Diego immediately after each other (in any order).**

How many ways can the trip be arranged now?

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

Realistically, choose order of visiting cities based on distance…
we wouldn't go to Los Angeles, then Minneapolis, then San Diego,
then New York, then Seattle, then Chicago, etc.

Must **start in New York** and **end in Seattle**.
Must also **visit Los Angeles and San Diego immediately after each other (in any order).**

How many ways can the trip be arranged now?

Traveling salesperson

Planning a trip to

New York
Chicago
Baltimore
Los Angeles
San Diego
Minneapolis
Seattle

Is there an order of visiting the cities that stops at each city exactly once and minimizes the distance traveled?

5! = 120

4,500

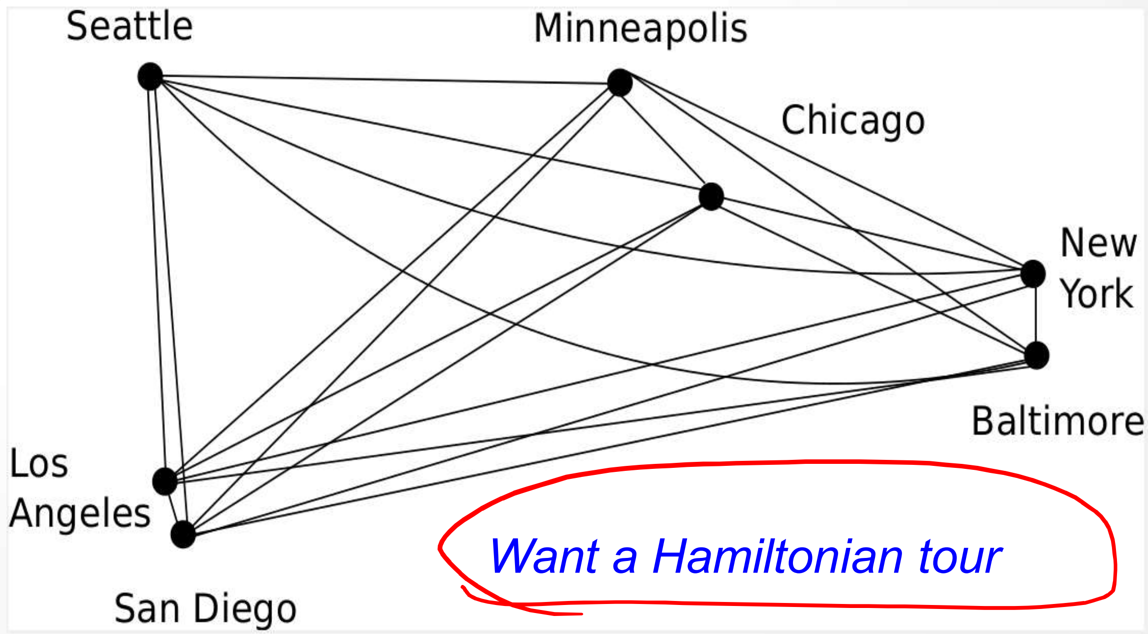
	NY	Chicago	Balt.	LA	SD	Minn.	Seattle
NY	0	800	200	2800	2800	1200	2900
Chicago	800	0	700	2000	2100	400	2000
Balt.	200	700	0	2600	2600	1100	2700
LA	2800	2000	2600	0	100	1900	1100
SD	2800	2100	2600	100	0	2000	1300
Minn.	1200	400	1100	1900	2000	0	1700
Seattle	2900	2000	2700	1100	1300	1700	0

Traveling salesperson

Planning a trip to

- New York
- Chicago
- Baltimore
- Los Angeles
- San Diego
- Minneapolis
- Seattle

Is there an order of visiting the cities that stops at each city exactly once and minimizes the distance traveled?



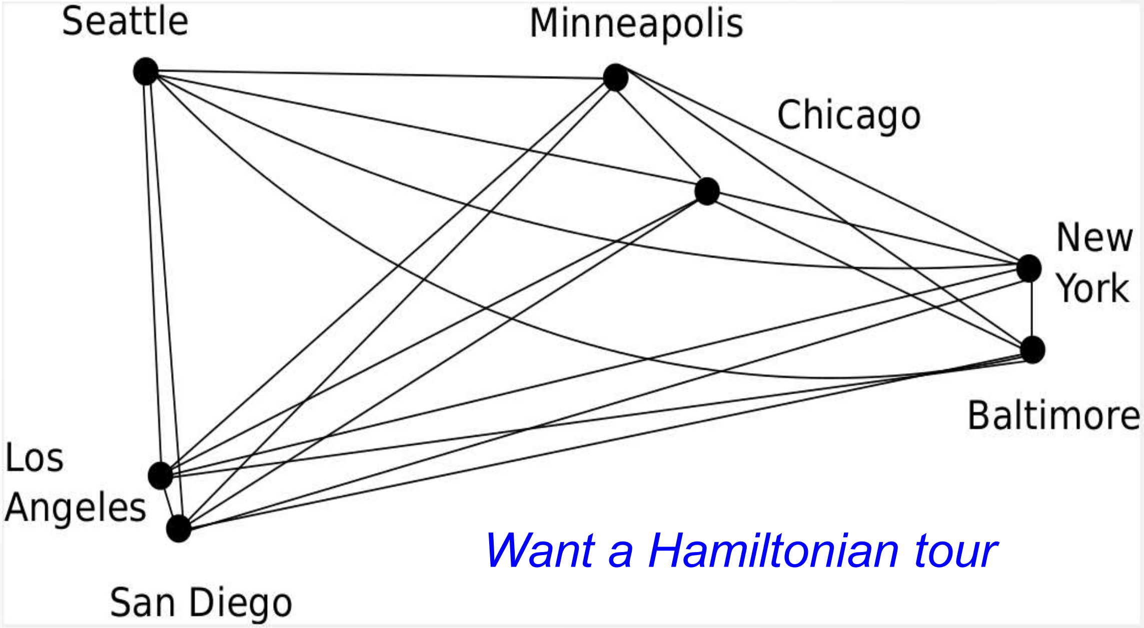
Want a Hamiltonian tour

Traveling salesperson

Developing an algorithm which, given a set of cities and distances between them, computes a shortest distance path between all of them is **NP-hard** (considered intractable, very hard).

Is there **any** algorithm for this question?

A. No, it's not possible.
B. Yes, it's just very slow.
C. ?

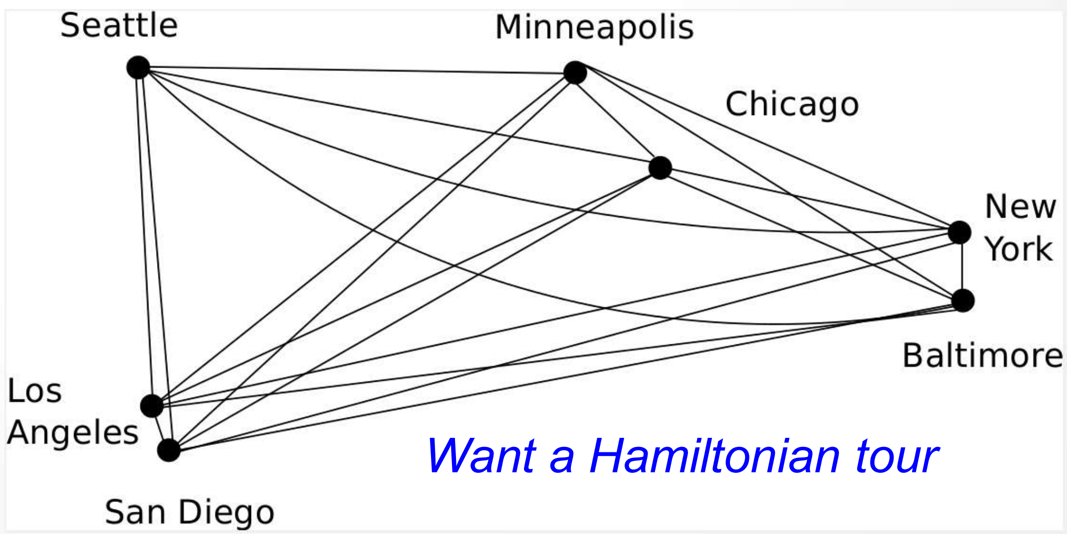


Traveling salesperson

Exhaustive search algorithm

List all possible orderings of the cities.
For each ordering, compute the distance traveled.
Choose the ordering with minimum distance.

How long does this take?



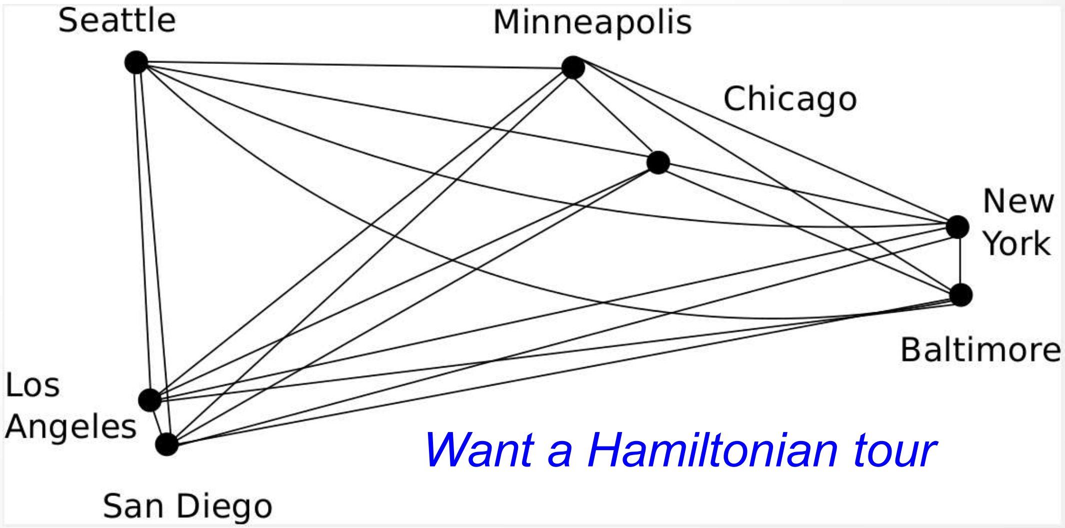
Traveling salesperson

Exhaustive search algorithm: given *n* cities and distances between them.

List all possible orderings of the cities.
For each ordering, compute the distance traveled.
Choose the ordering with minimum distance.

O(number of orderings)

How long does this take?



Traveling salesperson

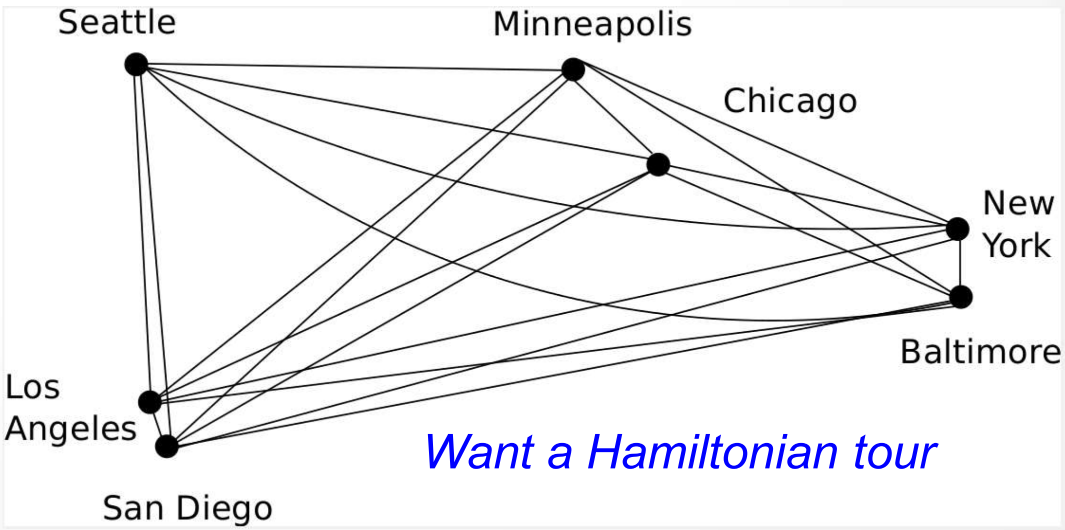
Exhaustive search algorithm: given *n* cities and distances between them.

List all possible orderings of the cities.
For each ordering, compute the distance traveled.
Choose the ordering with minimum distance.

O(number of orderings)

How long does this take?

A. O(n)
B. $O(n^2)$
C. $O(n^n)$
D. O(n!)
E. None of the above.



Traveling salesperson

Exhaustive search algorithm: given *n* cities and distances between them.

List all possible orderings of the cities.
For each ordering, compute the distance traveled.
Choose the ordering with minimum distance.

O(number of orderings)

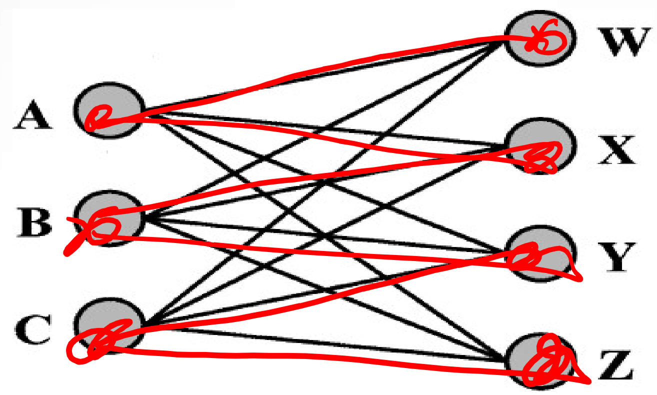
How long does this take?

A. O(n)
B. $O(n^2)$
C. $O(n^n)$
D. O(n!)
E. None of the above.

$2^n < n! < n^n$
for large n

Moral: counting gives upper bound on algorithm runtime.

Bipartite Graphs



Rosen p. 658

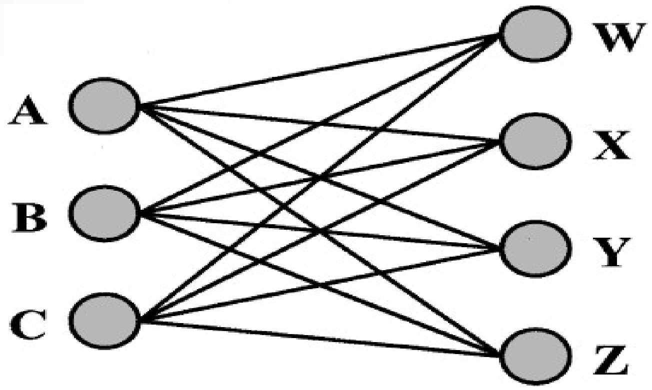
A **complete bipartite graph** is an undirected graph whose vertex set is partitioned into two sets V_1, V_2 such that

- there is an edge between each vertex in V_1 and each vertex in V_2
- there are no edges both of whose endpoints are in V_1
- there are no edges both of whose endpoints are in V_2

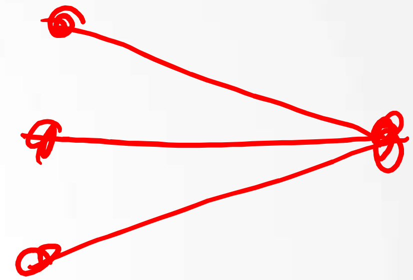
Is this graph Hamiltonian?

A. Yes
B. No

Bipartite Graphs



Rosen p. 658



A **complete bipartite graph** is an undirected graph whose vertex set is partitioned into two sets V_1, V_2 such that

- there is an edge between each vertex in V_1 and each vertex in V_2
- there are no edges both of whose endpoints are in V_1
- there are no edges both of whose endpoints are in V_2

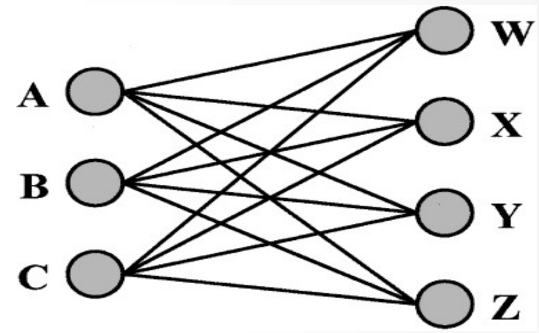
Is every complete bipartite graph Hamiltonian?

A. Yes
B. No

Bipartite Graphs

$$\underbrace{4\cdot3\cdot3\cdot2\cdot2\cdot1\cdot1}_{4!} \quad (3!)$$

Rosen p. 658



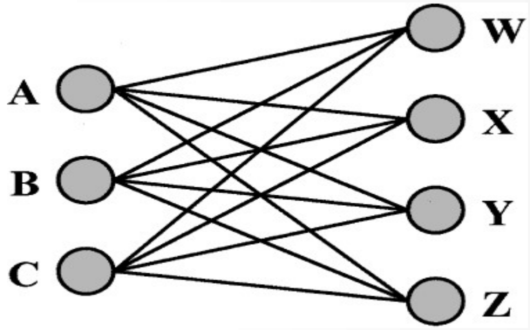
Claim: any **complete bipartite graph** with $|V_1| = k$, $|V_2| = k+1$ is Hamiltonian.

How many Hamiltonian tours can we find?

A. k
B. k(k+1)
C. k!(k+1)!
D. (k+1)!
E. None of the above.

Bipartite Graphs

Rosen p. 658



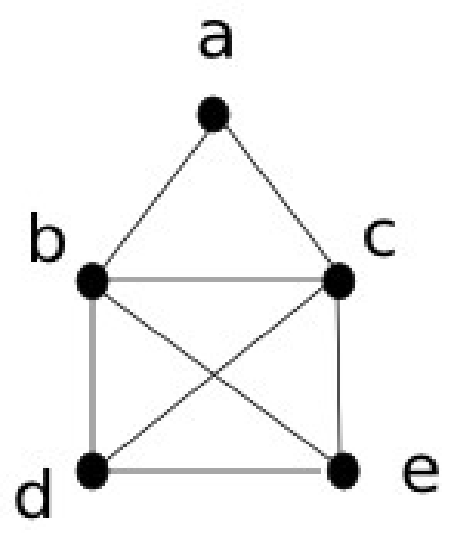
Claim: any **complete bipartite graph** with $|V_1| = k$, $|V_2| = k+1$ is Hamiltonian.

How many Hamiltonian tours can we find?

A. k
B. k(k+1)
C. k!(k+1)!
D. (k+1)!
E. None of the above.

Product rule!

When product rule fails

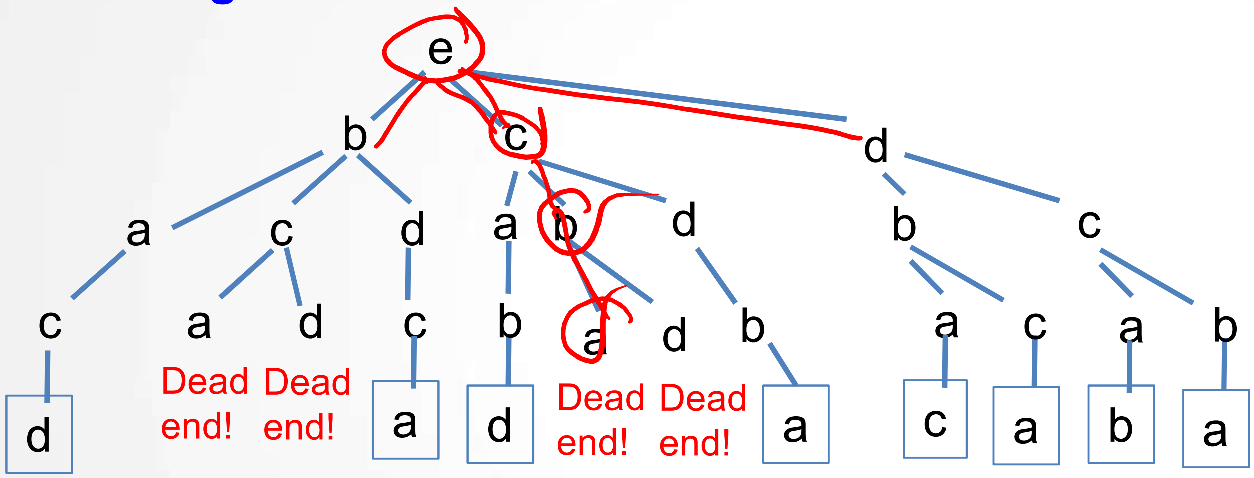


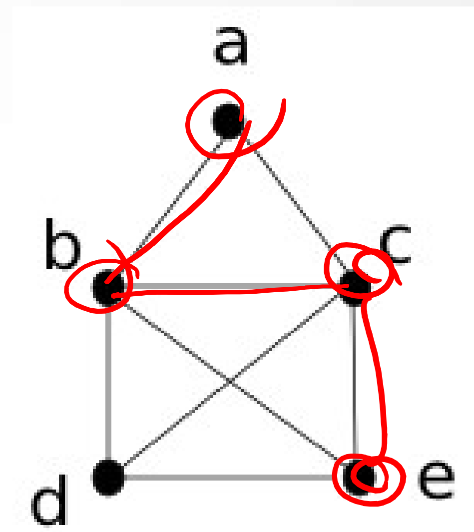
How many Hamiltonian tours can we find?

A. 5!
B. 5!4!
C. ?

When product rule fails

Tree Diagrams





Which Hamiltonian tours start at e?

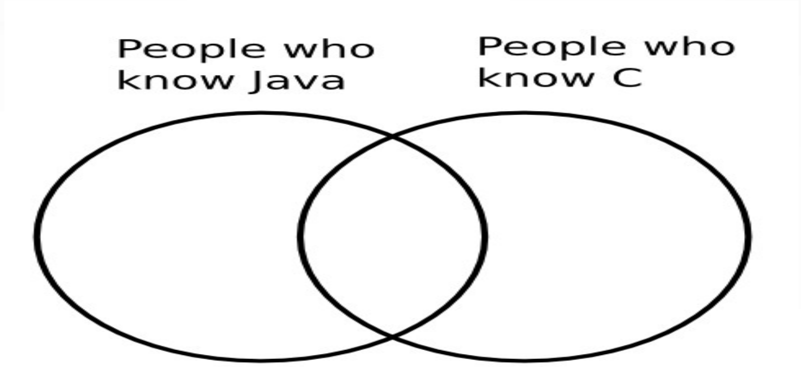
List all possible next moves.

Then count leaves.

Rosen p.394-395

When sum rule fails

Rosen p. 392-394



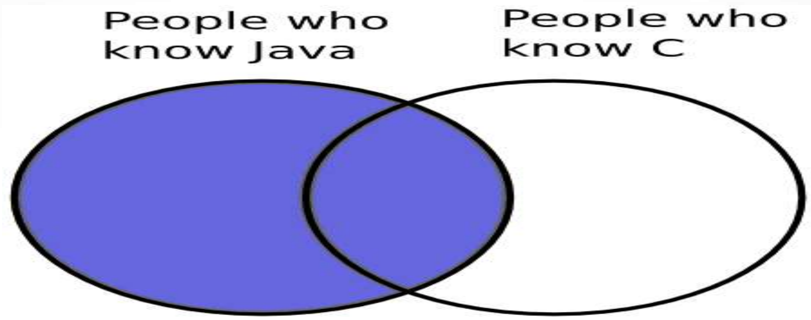
Let A = { people who know Java } and B = { people who know C }

How many people know Java or C (or both)?

A. $|A| + |B|$
B. $|A|\,|B|$
C. $|A|^{|B|}$
D. $|B|^{|A|}$
E. None of the above.

When sum rule fails

Rosen p. 392-394



Let A = { people who know Java } and B = { people who know C }

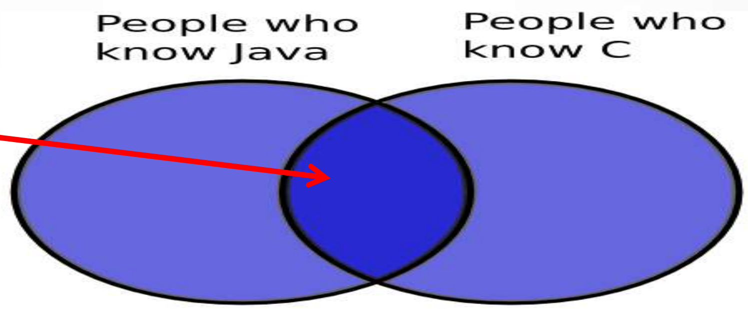
people who know Java or C = # people who know Java

$|A| + |B|$

When sum rule fails

Rosen p. 392-394

Double counted!



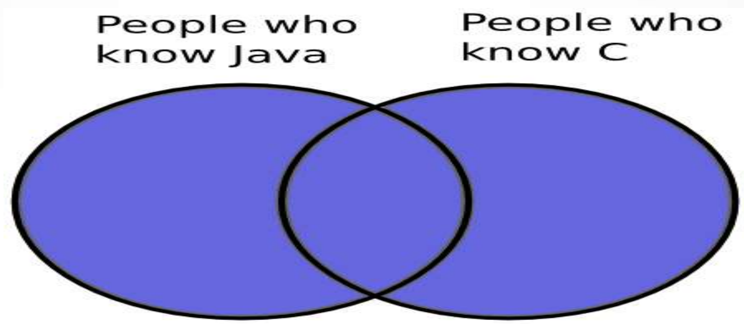
$|A| + |B|$

Let A = { people who know Java } and B = { people who know C }

\# people who know Java or C = # people who know Java
+ # people who know C

When sum rule fails

Rosen p. 392-394

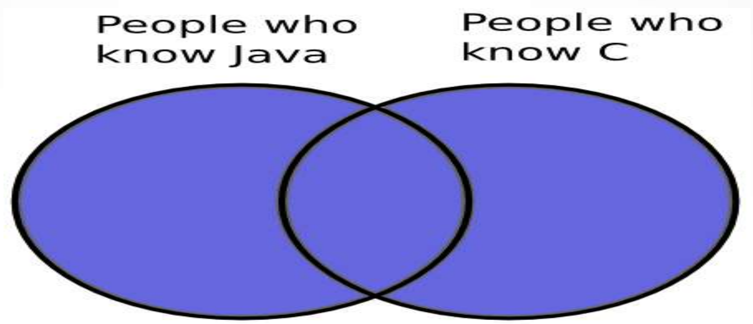


Let A = { people who know Java } and B = { people who know C }

people who know Java or C = # people who know Java
+ # people who know C
- # people who know both

Inclusion-Exclusion principle

Rosen p. 392-394

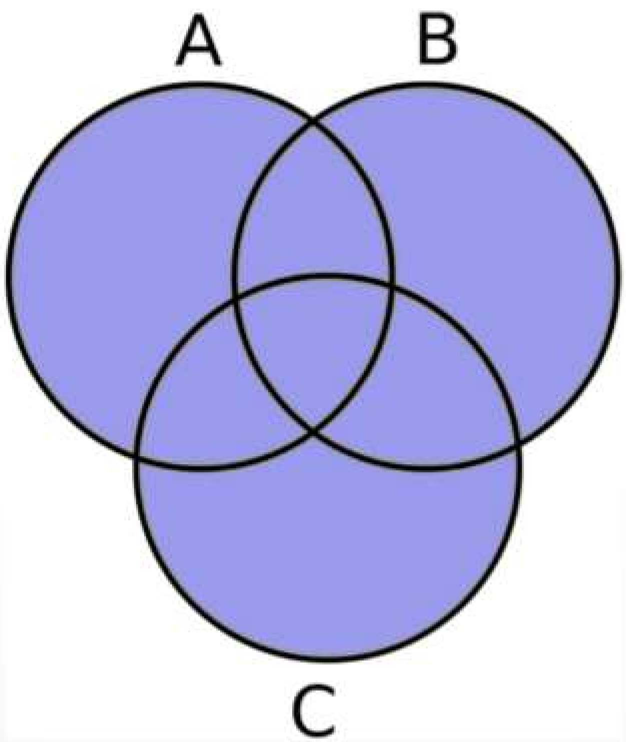


Let A = { people who know Java } and B = { people who know C }

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

Inclusion-Exclusion for three sets

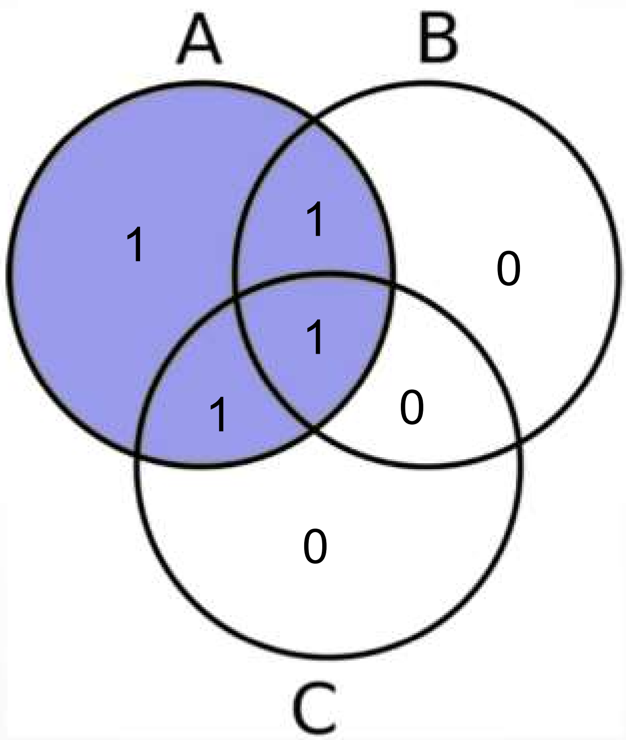
Rosen p. 392-394



$|A \cup B \cup C| = ?$

Inclusion-Exclusion for three sets

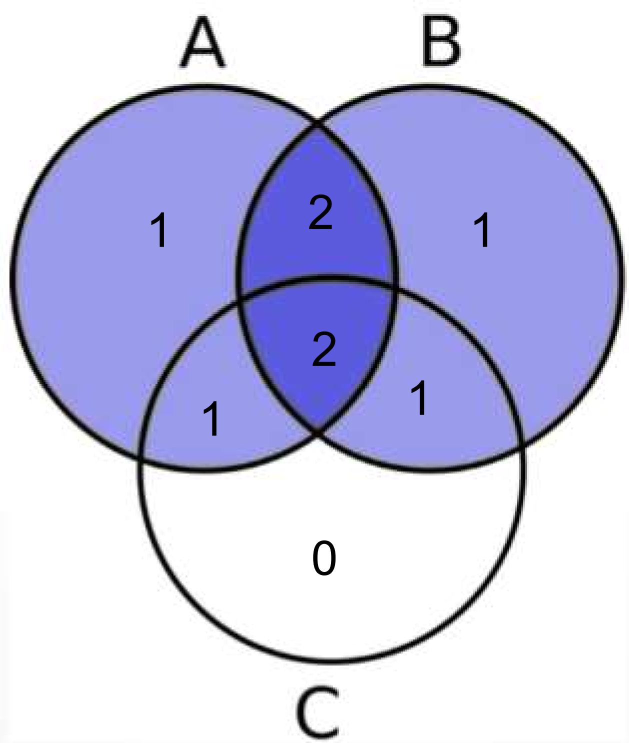
Rosen p. 392-394



$|A \cup B \cup C| = |A| + \cdots$

Inclusion-Exclusion for three sets

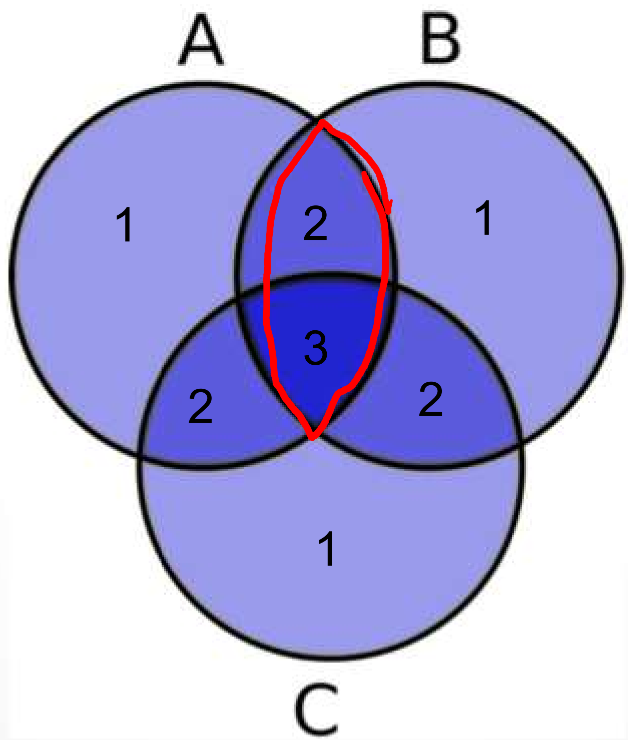
Rosen p. 392-394



$|A \cup B \cup C| = |A| + |B| + \cdots$

Inclusion-Exclusion for three sets

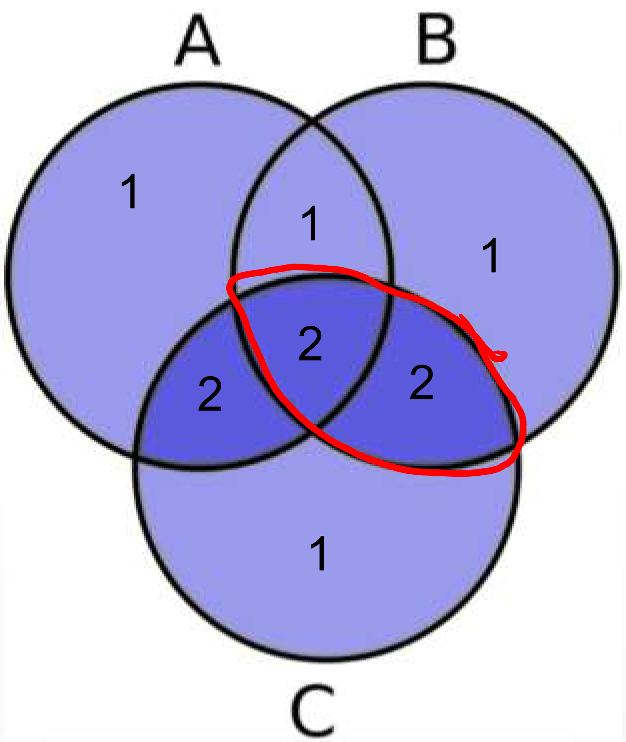
Rosen p. 392-394



$$|A \cup B \cup C| = |A| + |B| + |C| + \cdots$$

Inclusion-Exclusion for three sets

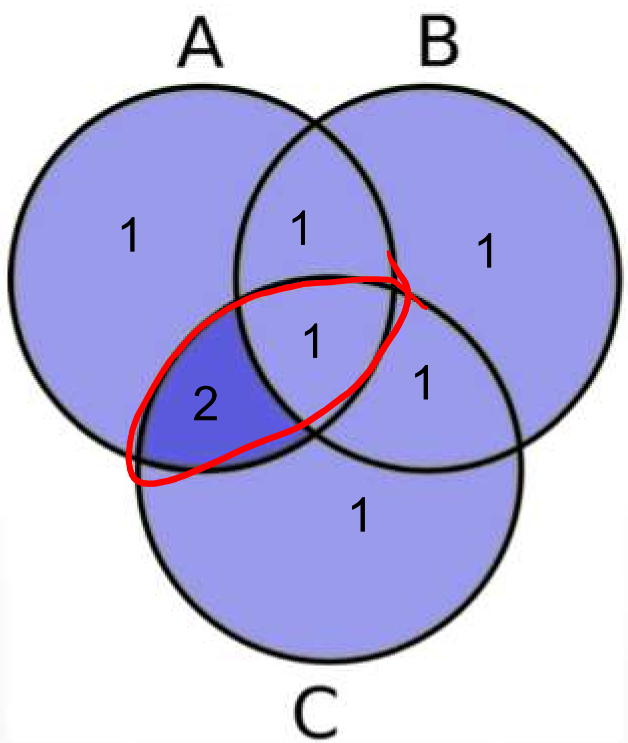
Rosen p. 392-394



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

Inclusion-Exclusion for three sets

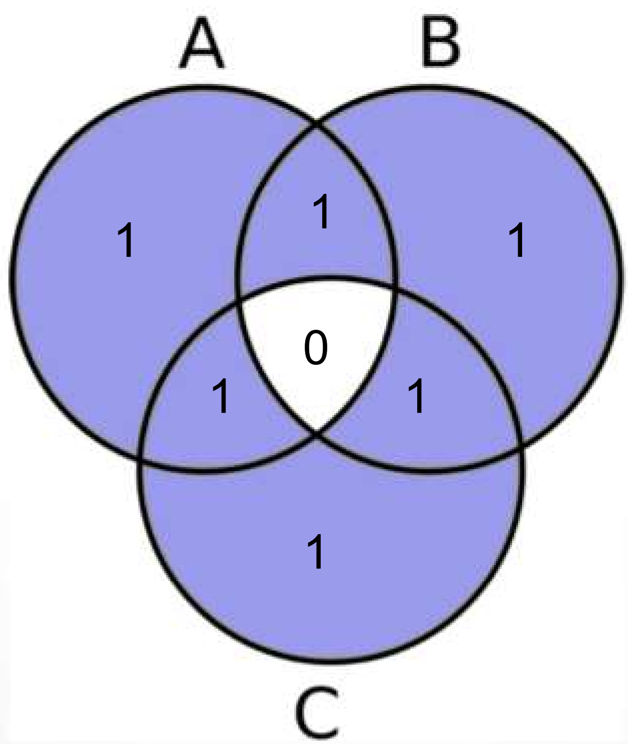
Rosen p. 392-394



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

Inclusion-Exclusion for three sets

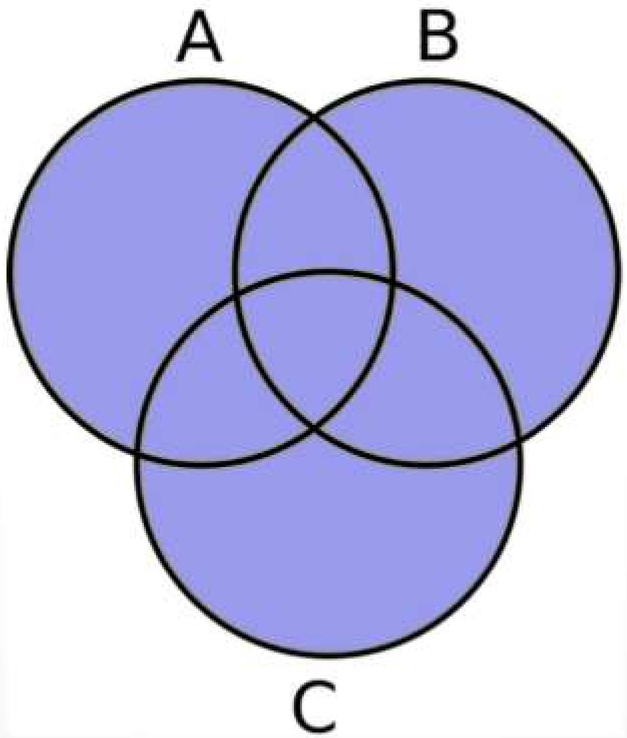
Rosen p. 392-394



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

Inclusion-Exclusion for three sets

Rosen p. 392-394



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

Inclusion-Exclusion principle

Rosen p. 556

If $A_1, A_2, \ldots, A_n$ are finite sets then

$$|A_1 \cup A_2 \cup \cdots A_n| = \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k|$$
$$- \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|$$

Templates

How many four-letter strings have one vowel and three consonants?

There are 5 vowels: AEIOU
and 21 consonants: BCDFGHJKLMNPQRSTVWXYZ.

A. $5*21^3$
B. 26^4
C. 5+52
D. $4*5*21^3$

5 21 21 21 x4

vowels c. c c

(0 1 0 0 0) (0 0 0 1 0 ...) (0 1 0 0)

E F

C V C C

Templates

How many four-letter strings have one vowel and three consonants?

There are 5 vowels: AEIOU
and 21 consonants: BCDFGHJKLMNPQRSTVWXYZ.

Template	*# Matching*
VCCC	5 * 21 * 21 * 21
CVCC	21 * 5 * 21 * 21
CCVC	21 * 21 * 5 * 21
CCCV	21 * 21 * 21 * 5

Total: $4*5*21^3$

Counting with categories

Rosen p. 394

If $A = X_1 \cup X_2 \cup \ldots \cup X_n$ and all X_i, X_j disjoint and all X_i have same size, then

$$|X_i| = |A| / n$$

More generally:

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, d of the n ways give the same result as w did.

Counting with categories

Rosen p. 394

If $A = X_1 \cup X_2 \cup \ldots \cup X_n$ and all X_i, X_j disjoint and all X_i have same size, then

$$|X_i| = |A| / n$$

More generally:

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, d of the n ways give the same result as w did.

Counting with categories

Rosen p. 394

If $A = X_1 \cup X_2 \cup \ldots \cup X_n$ and all X_i, X_j disjoint and all X_i have same size, then

$$|X_i| = |A| / n$$

Or in other words,

If objects are partitioned into categories of equal size, and we want to think of different objects as being the same if they are in the same category, then

categories = (# objects) / (size of each category)

Ice cream!

An ice cream parlor has n different flavors available. How many ways are there to order a two-scoop ice cream cup (where you specify which scoop goes on bottom and which on top, and the two flavors must be different)?

A. n^2
B. $n!$
C. $n(n-1)$
D. $2n$
E. None of the above.

Ice cream!

An ice cream parlor has n different flavors available.
How can we use our earlier answer to decide the number of cups for two scoops side by side,
if we count two cups as the same if they have the same two flavors?

A. Double the previous answer.
B. Divide the previous answer by 2.
C. Square the previous answer.
D. Keep the previous answer.
E. None of the above.

Ice cream!

An ice cream parlor has n different flavors available.
How can we use our earlier answer to decide the number of cups for two scoops side by side,
if we count two cups as the same if they have the same two flavors?

Objects: cones
Categories: (flavor pairs)
Size of each category: 2

different if each

categories = (# objects) / (size of each category)

(flavor pairs) = cones / 2
cups

Ice cream!

An ice cream parlor has n different flavors available.
How can we use our earlier answer to decide the number of cups for two scoops side by side,
if we count two cups as the same if they have the same two flavors?

Objects: ~~cups~~ cones
Categories: flavor pairs (regardless of order)
Size of each category:

categories = (# objects) / (size of each category)

Ice cream!

An ice cream parlor has n different flavors available.
How can we use our earlier answer to decide the number of cups for two scoops side by side,
if we count two cups as the same if they have the same two flavors?

Objects: ~~cups~~ Cones **n(n-1)**
Categories: flavor pairs (regardless of order) cups
Size of each category: **2**

categories = (n)(n-1)/ 2

Avoiding double-counting

Object Symmetries

How many different colored triangles can we create by tying these three pipe cleaners end-to-end?

A. $3!$
B. 2^3
C. 3^2
D. 1
E. None of the above.

Object Symmetries

How many different colored triangles can we create by tying these three pipe cleaners end-to-end?

3!

Objects: all different colored triangles
Categories: physical colored triangles (two triangles are the same if they can be rotated and/or flipped to look alike)
3 rotations × 2 reflections
Size of each category:

categories = (# objects) / (size of each category)

$1 = 3!/3 \cdot 2$

Object Symmetries

How many different colored triangles can we create by tying these three pipe cleaners end-to-end?

Objects: all different colored triangles **3!**
Categories: physical colored triangles (two triangles are the same if they can be rotated and/or flipped to look alike)
Size of each category: **(3)(2)** three possible rotations, two possible flips

categories = (# objects) / (size of each category) = **6/6 = 1**

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

Density is number of ones

For example, n=6 k=4

Which of these strings matches this example?

A. 101101
B. 1100011101
C. 111011
D. 1101
E. None of the above.

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

Density is number of ones

For example, n=6 k=4

Product rule: How many options for the first bit? the second? the third?

$2 \cdot 2 \cdot 2 \cdot \ldots$

Fixed-density Binary Strings

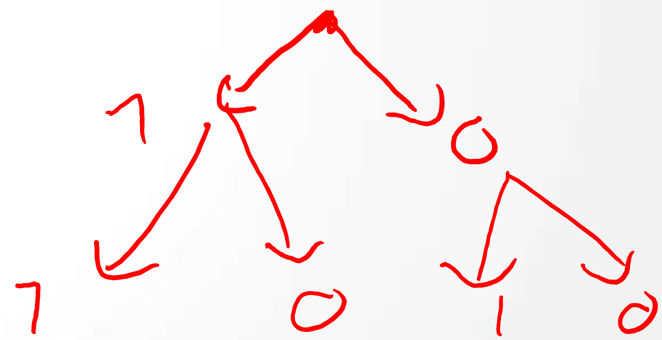
Rosen p. 413

How many length n **binary strings contain** k **ones**?

Density is number of ones

For example, n=6 k=4

Tree diagram: *gets very big & is hard to generalize*



Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

Density is number of ones

For example, n=6 k=4

Another approach: **use a different representation** i.e. count with categories

Objects:
Categories:
Size of each category:

categories = (# objects) / (size of each category)

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

For example, n=6 k=4

Another approach: **use a different representation** i.e. count with categories

Objects: all strings made up of 0_1, 0_2, 1_1, 1_2, 1_3, 1_4
Categories: strings that agree except subscripts
Size of each category:

Subscripts so objects are distinct

categories = (# objects) / (size of each category)

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

For example, n=6 k=4

Another approach: **use a different representation** i.e. count with categories

Objects: all strings made up of $0_1, 0_2, 1_1, 1_2, 1_3, 1_4$ **6!**

Categories: strings that agree except subscripts

Size of each category: **?** $4! \cdot 2!$

$n!$

$(n-k)!\,k!$

\# categories = (# objects) / (size of each category)

Fixed-density Binary Strings

How many subscripted strings i.e. rearrangements of the symbols

$$0_1, 0_2, 1_1, 1_2, 1_3, 1_4$$

result in

$$101101$$

when the subscripts are removed?

A. 6!
B. 4!
C. 2!
D. 4!2!
E. None of the above

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

For example, n=6 k=4

Another approach: **use a different representation** i.e. count with categories

Objects: all strings made up of 0_1, 0_2, 1_1, 1_2, 1_3, 1_4 **6!**
Categories: strings that agree except subscripts
Size of each category: **4!2!**

\# categories = (# objects) / (size of each category)
= 6! / (4!2!)

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

Another approach: **use a different representation** i.e. count with categories

Objects: all strings made up of $0_1, 0_2, \ldots, 0_{n-k}, 1_1, 1_2, \ldots, 1_k$ **n!**
Categories: strings that agree except subscripts
Size of each category: **k!(n-k)!**

 # categories = (# objects) / (size of each category)
= n!/ (k! (n-k) !)

Terminology

Rosen p. 407-413

A **permutation** of r elements from a set of n *distinct* objects is an **ordered** arrangement of them. There are

P(n,r) = n(n-1) (n-2) …(n-r+1)

many of these.

A **combination** of r elements from a set of n *distinct* objects is an **unordered** selection of them. There are

C(n,r) = n!/ (r! (n-r) !)

many of these.

Binomial coefficient
"n choose r" $\binom{n}{r}$

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

How to express this using the new terminology?

A. C(n,k)
B. C(n,n-k)
C. P(n,k)
D. P(n,n-k)
E. None of the above

Fixed-density Binary Strings

Rosen p. 413

How many length n **binary strings contain** k **ones**?

How to express this using the new terminology?

A. C(n,k) {1,2,3..n} is set of positions in string, choose k positions for 1s
B. C(n,n-k) {1,2,3..n} is set of positions in string, choose n-k positions for 0s
C. P(n,k)
D. P(n,n-k)
E. None of the above

Ice cream! redux

An ice cream parlor has n different flavors available.

How many ice cream cones are there, if we count two cones as the same if they have the same two flavors (even if they're in opposite order)?

Objects: cones **n(n-1)**
Categories: flavor pairs (regardless of order)
Size of each category: **2**

categories = (n)(n-1)/ 2

Order doesn't matter so selecting a subset of size 2 of the n possible flavors:

$C(n,2) = n!/ (2! (n-2)!) = n(n-1)/2$

What's in a name?

Rosen p. 415

Binomial: sum of two terms, say x and y.
What do powers of binomials look like?

$$
\begin{aligned}
(x+y)^4 &= (x+y)(x+y)(x+y)(x+y) \\
&= (x^2+2xy+y^2)(x^2+2xy+y^2) \\
&= x^4+4x^3y+6x^2y^2+4xy^3+y^4
\end{aligned}
$$

In general , for $(x+y)^n$

A. All terms in the expansion are (some coefficient times) $x^k y^{n-k}$ for some k, 0<=k<=n.
B. All coefficients in the expansion are integers between 1 and n.
C. There is symmetry in the coefficients in the expansion.
D. The coefficients of x^n and y^n are both 1.
E. All of the above.

Binomial Theorem

Rosen p. 416

$(x+y)^n \quad = \quad (x+y)(x+y)\ldots(x+y)$

$= \quad x^n + \_\_ x^{n-1}y + \_\_ x^{n-2}y^2 + \ldots + \_\_ x^k y^{n-k} + \ldots + \_\_ x^2 y^{n-2} + \_\_ xy^{n-1} + y^n$

Number of ways we can choose k of the n factors (to contribute to x)
and hence also n-k of the factors (to contribute to y)

Binomial Theorem

Rosen p. 416

$$(x+y)^n = (x+y)(x+y)\ldots(x+y)$$

$$= x^n + \_\_\, x^{n-1}y + \_\_\, x^{n-2}y^2 + \ldots + \_\_\, x^k y^{n-k} + \ldots + \_\_\, x^2 y^{n-2} + \_\_\, xy^{n-1} + y^n$$

Number of ways we can choose k of the n factors (to contribute to x) and hence also n-k of the factors (to contribute to y) C(n,k)

$$= x^n + C(n,1)\, x^{n-1}y + \ldots + C(n,k)\, x^k y^{n-k} + \ldots + C(n,k-1)\, xy^{n-1} + y^n$$

Binomial Coefficient Identities

What's an **identity** ? An equation that is always true.

To prove

LHS = RHS

- Use algebraic manipulations of formulas

OR

- Interpret each side as counting some collection of strings, and then prove a statements about those sets of strings

Symmetry Identity

Rosen p. 411

Theorem: $\binom{n}{k} = \binom{n}{n-k}$

Symmetry Identity

Rosen p. 411

Theorem: $$\binom{n}{k} = \binom{n}{n-k}$$

Proof 1: Use formula $$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!k!} = \binom{n}{n-k}$$

Symmetry Identity

Rosen p. 411

Theorem: $$\binom{n}{k} = \binom{n}{n-k}$$

Proof 1: Use formula $$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!k!} = \binom{n}{n-k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n with k ones
RHS counts number of binary strings of length n with n-k ones

Symmetry Identity

Rosen p. 411

Theorem: $\binom{n}{k} = \binom{n}{n-k}$

Proof 1: Use formula $$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!k!} = \binom{n}{n-k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n with k ones and n-k zeros
RHS counts number of binary strings of length n with n-k ones and k zeros

Symmetry Identity

Rosen p. 411

Theorem: $$\binom{n}{k} = \binom{n}{n-k}$$

Proof 1: Use formula $$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!k!} = \binom{n}{n-k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n with k ones and n-k zeros
RHS counts number of binary strings of length n with n-k ones and k zeros

Can match up these two sets by pairing each string with another where 0s, 1s are flipped. This ***bijection*** *means the two sets have the same size. So* **LHS** = **RHS**.

Pascal's Identity

Rosen p. 418

Theorem: $$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Proof 1: Use formula

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings ???
RHS counts number of binary strings ???

Pascal's Identity

Rosen p. 418

Theorem: $$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n+1 that have k ones.
RHS counts number of binary strings ???

Length n+1 binary strings with k ones

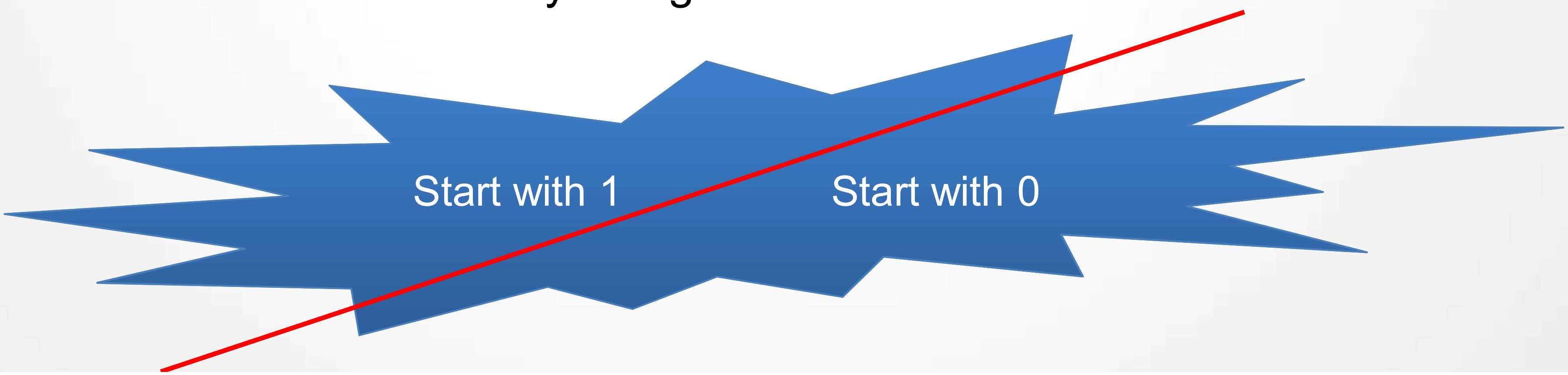
Pascal's Identity

Rosen p. 418

Theorem: $$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n+1 that have k ones.
RHS counts number of binary strings ???



Pascal's Identity

Rosen p. 418

How many length n+1 strings start with 1 and have k ones in total?

A. C(n+1, k+1)
B. C(n, k)
C. C(n, k+1)
D. C(n, k-1)
E. None of the above.

Pascal's Identity

Rosen p. 418

How many length n+1 strings start with 0 and have k ones in total?

A. C(n+1, k+1)
B. C(n, k)
C. C(n, k+1)
D. C(n, k-1)
E. None of the above.

Start with 1 Start with 0

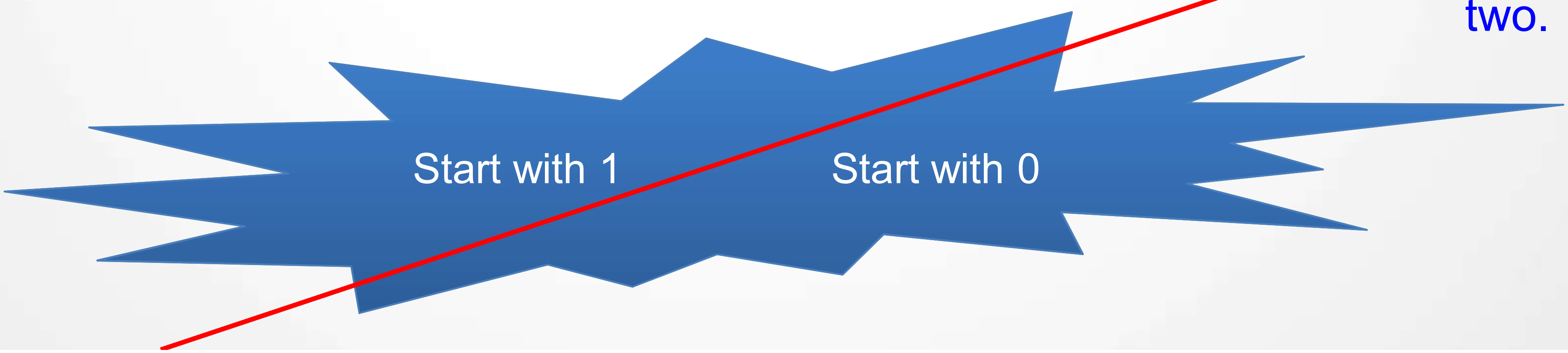
Pascal's Identity

Rosen p. 418

Theorem: $$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Proof 2: Combinatorial interpretation?

LHS counts number of binary strings of length n+1 that have k ones.
RHS counts number of binary strings of length n+1 that have k ones, split into two.



Sum Identity

Rosen p. 417

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

What set does the **LHS** count?

A. Binary strings of length n that have k ones.
B. Binary strings of length n that start with 1.
C. Binary strings of length n that have any number of ones.
D. None of the above.

Sum Identity

Rosen p. 417

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

Proof : Combinatorial interpretation?

LHS counts number of binary strings of length n that have any number of 1s.

By sum rule, we can break up the set of binary strings of length n into disjoint sets based on how many 1s they have, then add their sizes.

RHS counts number of binary strings of length n.

This is the same set so **LHS** = **RHS**.