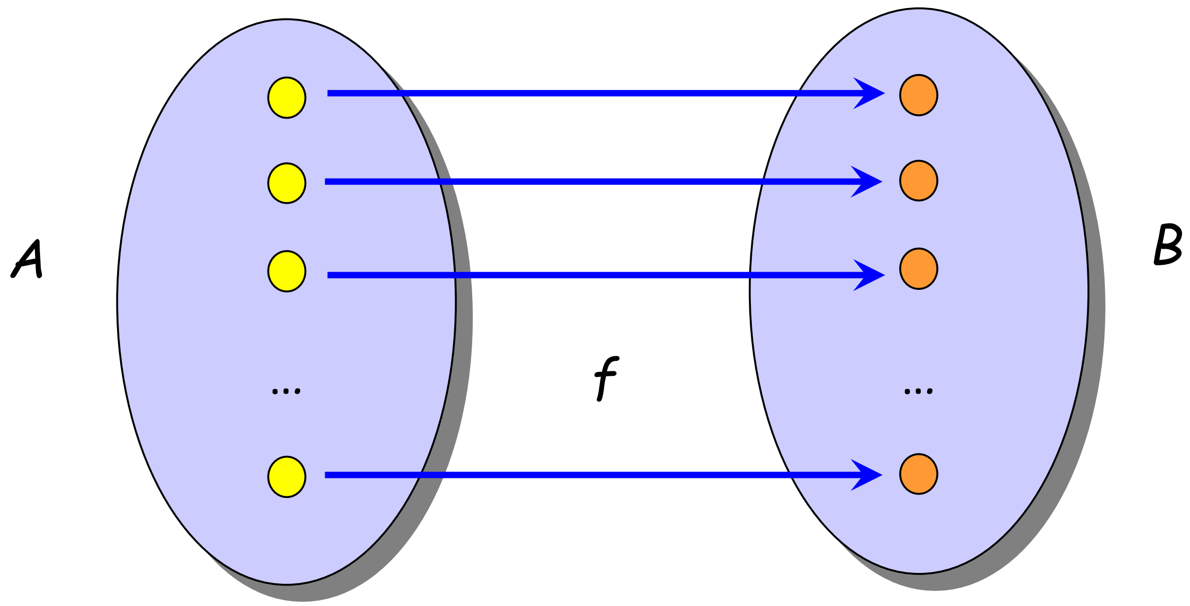
Counting by Mapping



Plan

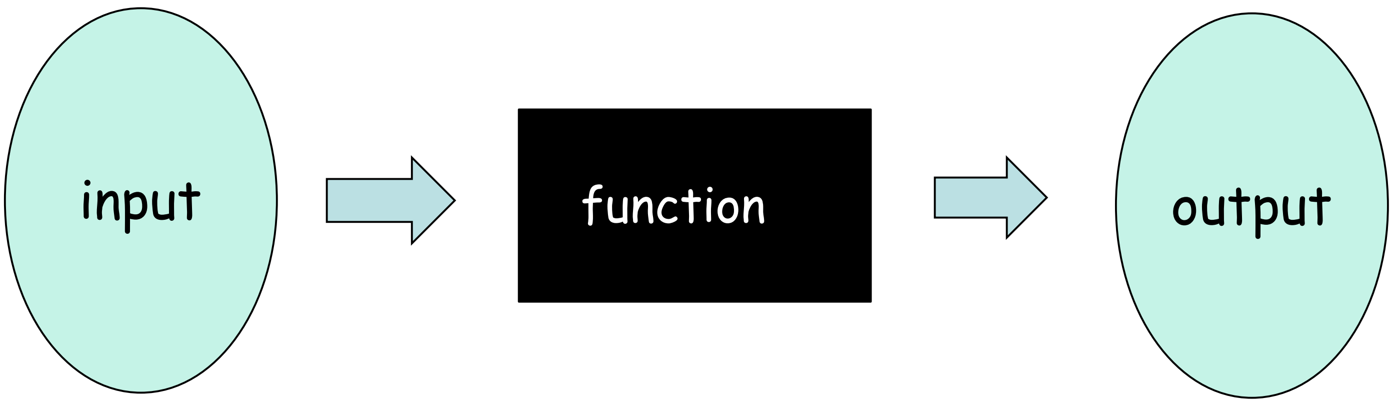
We will study how to define mappings to count.

There will be many examples shown.

- Functions
- Bijection rule
- Division rule
- Catalan number

Functions

Informally, we are given an "input set",
and a function which gives an output for each possible input.



The important point is that there is only one output for each input.

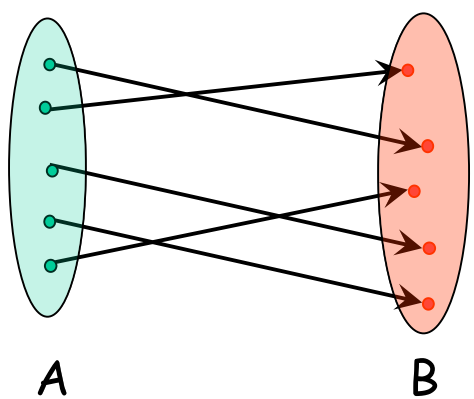
We say a function f "maps" the elements of an input set A to the elements of an output set B.

Functions

More formally, we write $f : A \rightarrow B$

to represent that f is a function from set A to set B, which

associates each $a \in A$ with an element $f(a) \in B$.



The *domain (input)* of f is A.

The *range (output)* of f is $f(A)$.

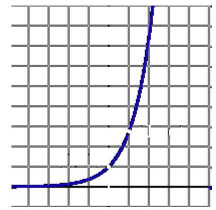
The *codomain* of f is $B \supseteq f(A)$.

For each input there is exactly one output.

Note: the input set can be the same as the output set, e.g. both are integers.

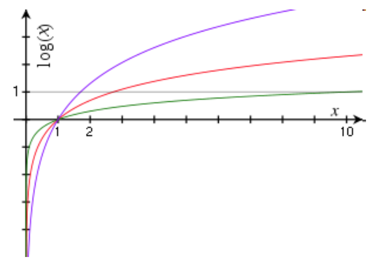
Examples of Functions

$$f(x) = e^x$$



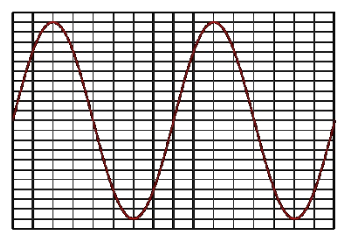
domain = $\mathbb{R}$
range = $\mathbb{R}^+$

$$f(x) = \log(x)$$



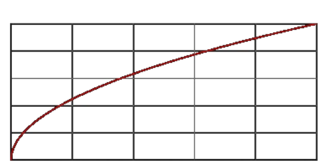
domain = $\mathbb{R}^+$
range = $\mathbb{R}$

$$f(x) = \sin(x)$$



domain = $\mathbb{R}$
range = [-1,1]

$$f(x) = \sqrt{x}$$



domain = $\{x \in \mathbb{R} \mid x>=0 \}$
range = $\{x \in \mathbb{R} \mid x>=0 \}$

Examples of Functions

$f(S) = |S|$

domain = the set of all finite sets
range = non-negative integers

f(string) = Length(string)

domain = the set of all finite strings
range = non-negative integers

f(student-name) = student-ID

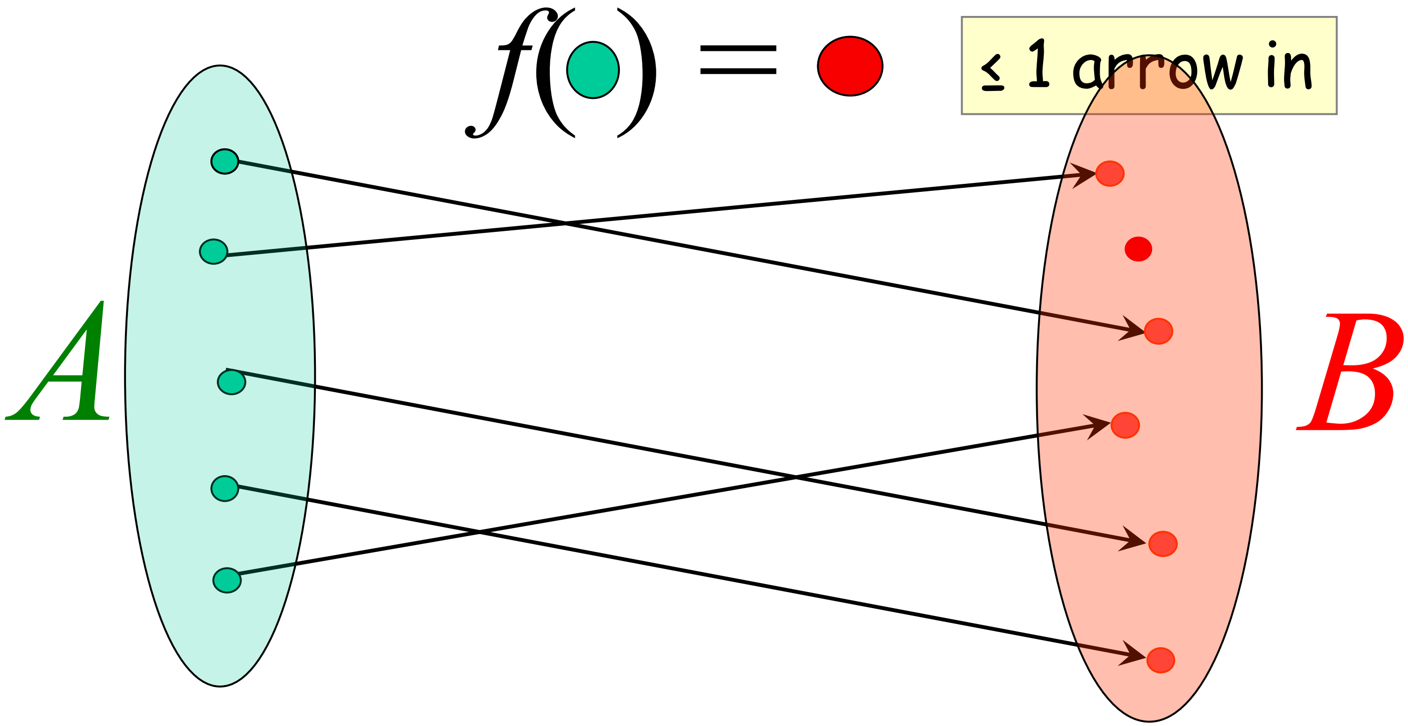
not a function,
since the same name may
corresponds to different students

f(x) = Is-Prime(x)

domain = positive integers
range = {T,F}

Injections

$f : A \to B$ is injective if no two inputs have the same output.

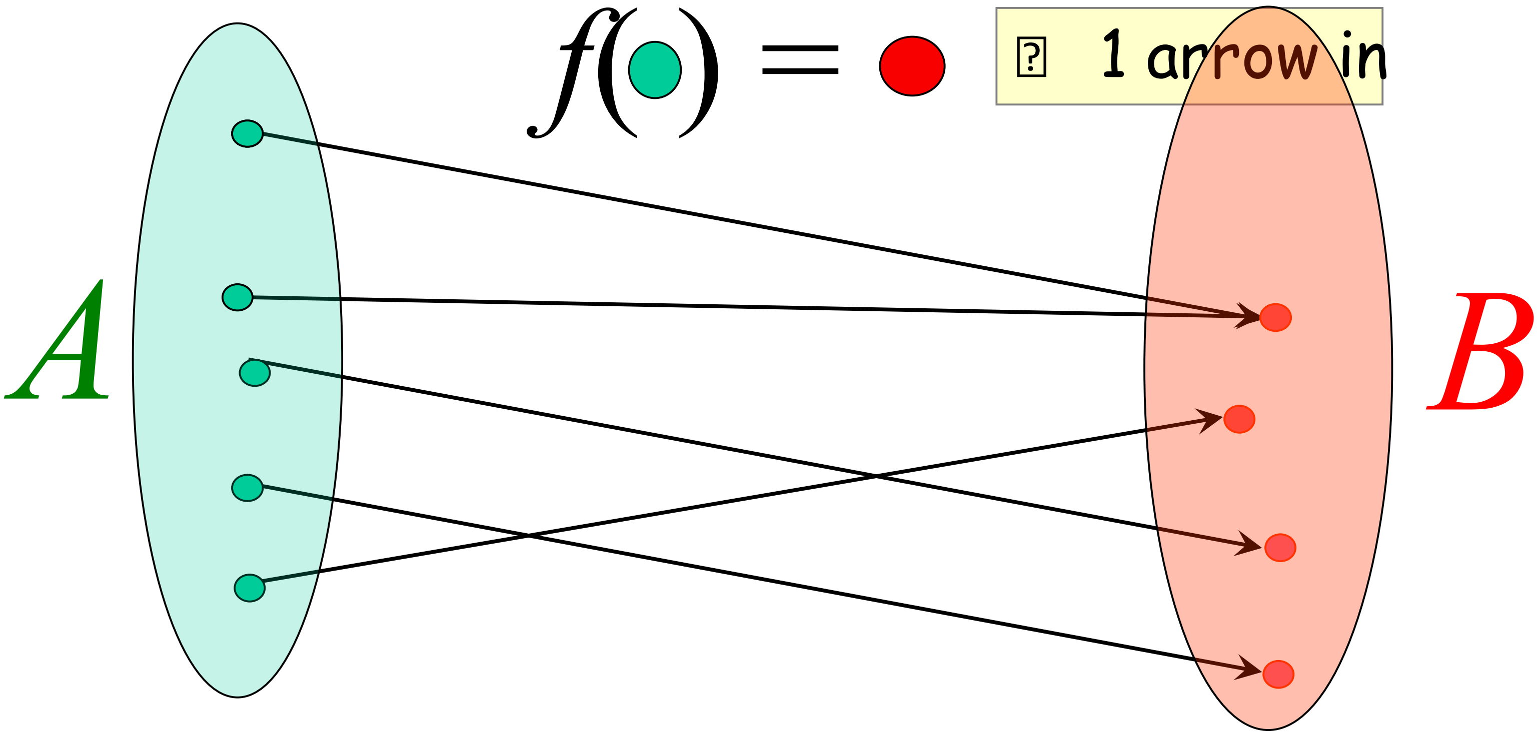


$$\boxed{|A| \leq |B|}$$

Surjections

$f : A \to B$ is surjective if every output is possible.

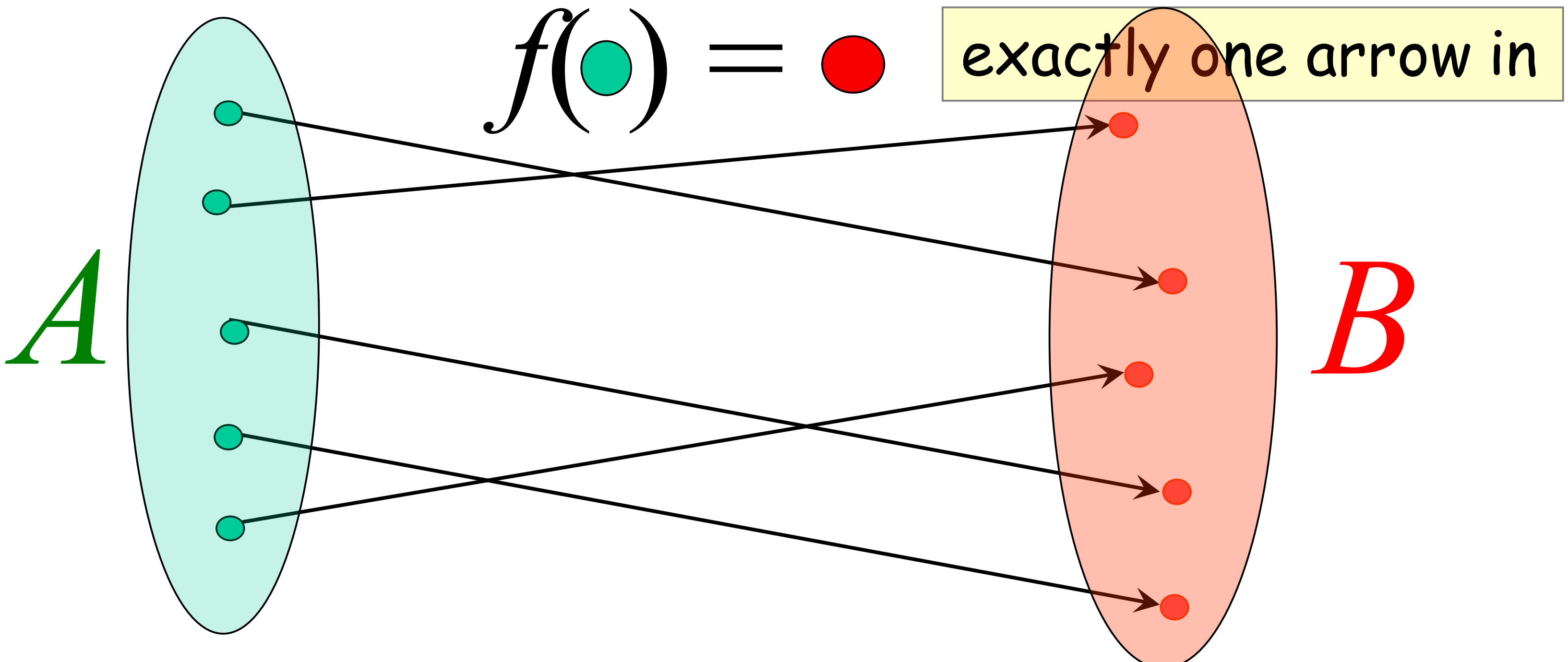
$f(\bullet) = \bullet$ ≥ 1 arrow in



$$|A| \geq |B|$$

Bijections

$f : A \to B$ is bijective if it is both surjective and injective.



$$|A| = |B|$$

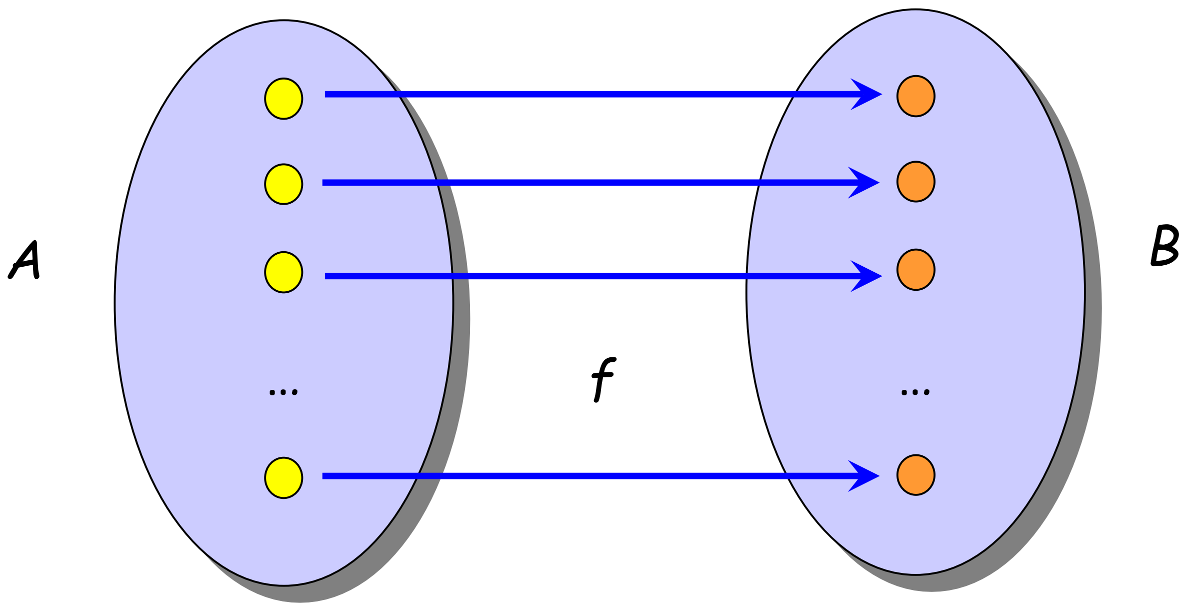
This Lecture

- Functions
- Bijection rule
- Division rule
- Catalan number

Counting Rule: Bijection

If f is a bijection from A to B,

then $|A| = |B|$.



To compute $|A|$, one strategy is to define a bijection between A and B, where B is easier to count and we can compute $|B|$ instead.

Power Set

Reall that

$Pow(S)$ = the power set of S

= the set of all subsets of S

for $S = \{a, b, c\}$,

$Pow(S) = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$

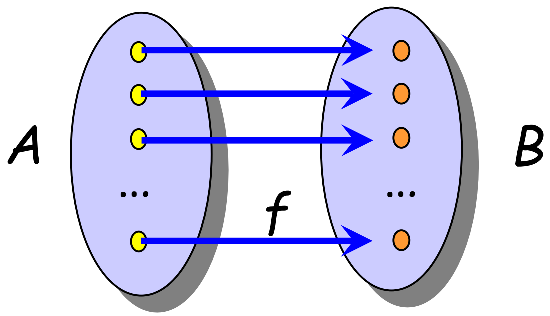
Suppose S has n elements.

How to count |Pow(S)|?

Bijection: Power Set and Binary Strings

S: $\{s_1, s_2, s_3, s_4, s_5, \ldots, s_n\}$

We define a bijection f between subsets and binary strings



A: the set of all subsets of S

B: the set of all n-bit strings

The mapping f: $A \rightarrow B$ is defined in the following way:

Given a subset $T \subseteq S$, we define $f(T)$ as an n-bit string such that the i-th bit equal to 1 if and only if $s_i \in T$.

Bijection: Power Set and Binary Strings

S: {a,b,c}

Pow(S) → 3-bit strings

$\emptyset, \longrightarrow 000$

$\{a\}, \longrightarrow 100$

$\{b\}, \longrightarrow 010$

$\{c\}, \longrightarrow 001$

$\{a,b\}, \longrightarrow 110$

$\{a,c\}, \longrightarrow 101$

$\{b,c\}, \longrightarrow 011$

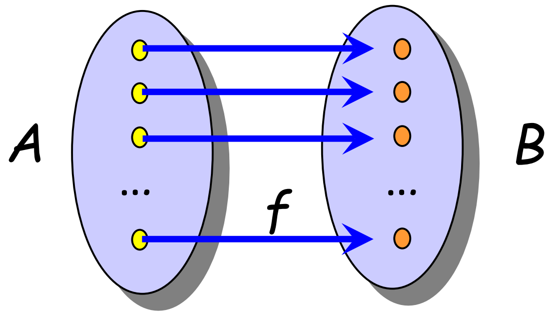
$\{a,b,c\} \longrightarrow 111$

Bijection: Power Set and Binary Strings

The mapping is defined in the following way:

subset: $\{s_1, \quad s_3, s_4, \quad \ldots \ , s_n\}$

string: $1 \quad 0 \quad 1 \quad 1 \quad 0 \quad \ldots \quad 1$



This mapping is a bijection, because

- two different subsets are mapped to two different strings (injection)
- each binary string is mapped by some subset (surjection).

Therefore, |A| = |B|, where |B| can be computed directly.

So, $|Pow(S)| = |n\text{-bit binary strings}| = 2^n$.

Poker Hands

There are 52 cards in a deck.

Each card has a suit and a value.

4 suits (♠ ♥ ♦ ♣)

13 values (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A)

Five-Card Draw is a card game in which each player is initially dealt a hand, a subset of 5 cards.

How many different hands?

$$\binom{52}{5} = 2598960$$

Bijection: Full House

A Full House is a hand with three cards of one value and two cards of another value.

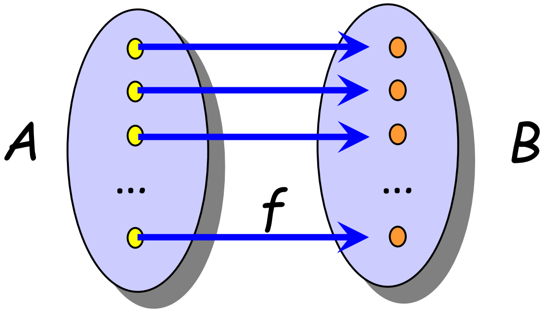
There is a **bijection** between Full Houses and sequences specifying:

1. The value of the triple, which can be chosen in 13 ways.
2. The suits of the triple, which can be selected in $\binom{4}{3}$ ways.
3. The value of the pair, which can be chosen in 12 ways.
4. The suits of the pair, which can be selected in $\binom{4}{2}$ ways.

$$\{\ 2\spadesuit,\ 2\clubsuit,\ 2\diamondsuit,\ J\clubsuit,\ J\diamondsuit\ \} \leftrightarrow (2, \{\spadesuit, \clubsuit, \diamondsuit\}, J, \{\clubsuit, \diamondsuit\})$$
$$\{\ 5\diamondsuit,\ 5\clubsuit,\ 5\heartsuit,\ 7\heartsuit,\ 7\clubsuit\ \} \leftrightarrow (5, \{\diamondsuit, \clubsuit, \heartsuit\}, 7, \{\heartsuit, \clubsuit\})$$

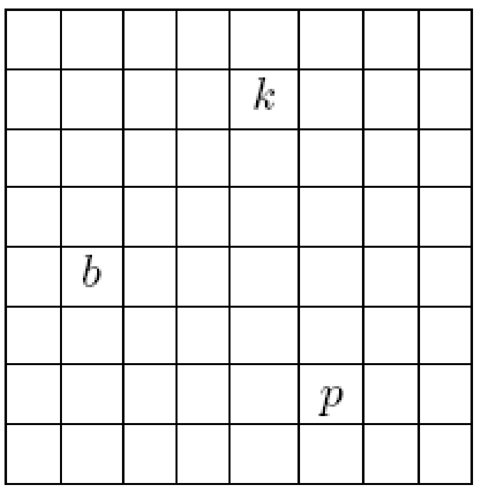
A: the set of full houses

B: the set of sequences which satisfy (1)-(4).

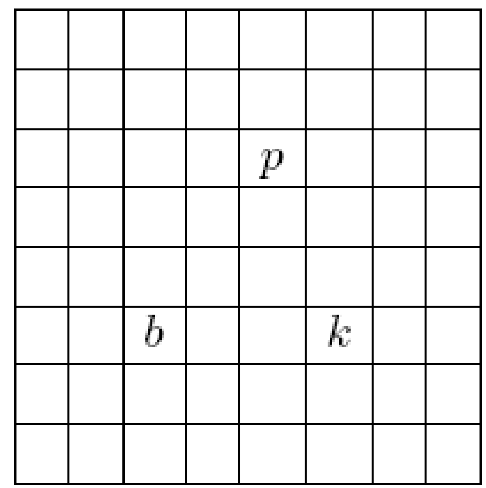


A Chess Problem

In how many different ways can we place a pawn (p), a knight (k), and a bishop (b) on a chessboard so that no two pieces share a row or a column?



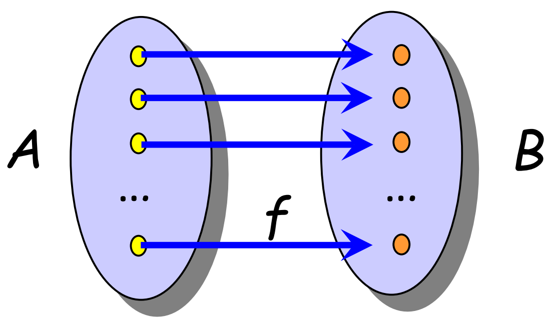
valid



invalid

A Chess Problem

We define a mapping f between configurations to sequences (r(p), c(p), r(k), c(k), r(b), c(b)), where r(p), r(k), and r(b) are distinct rows, and c(p), c(k), and c(b) are distinct columns.



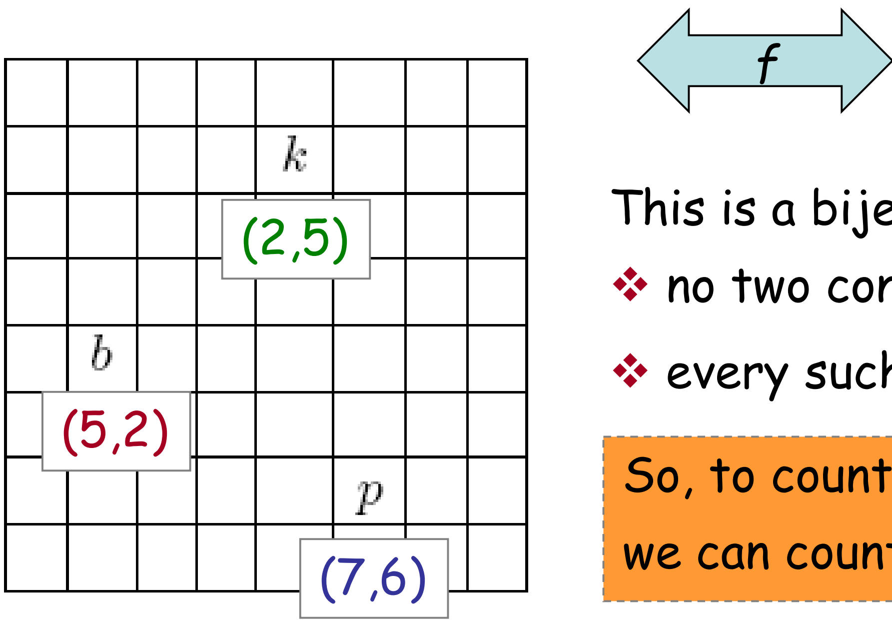
A = {the configurations of the 3 pieces}

B = {restricted sequences of 6 numbers}

If we can define a bijection between A and B, and also calculate |B|, then we can determine |A|.

A Chess Problem

We define a mapping f between configurations to sequences (r(p), c(p), r(k), c(k), r(b), c(b)), where r(p), r(k), and r(b) are distinct rows, and c(p), c(k), and c(b) are distinct columns.



(7,6,2,5,5,2)

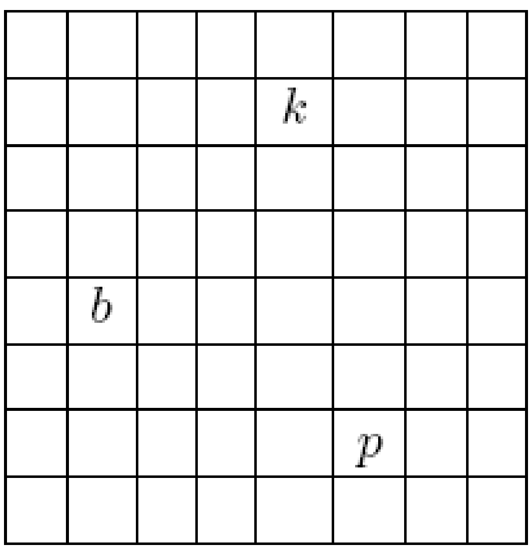
This is a bijection, because:

- no two configs map to same sequence (injection)
- every such sequence is mapped (surjection)

So, to count the number of chess configurations, we can count the restricted 6-sequences instead.

A Chess Problem

We define a mapping f between configurations to sequences (r(p), c(p), r(k), c(k), r(b), c(b)), where r(p), r(k), and r(b) are distinct rows, and c(p), c(k), and c(b) are distinct columns.



(7,6,2,5,5,2)

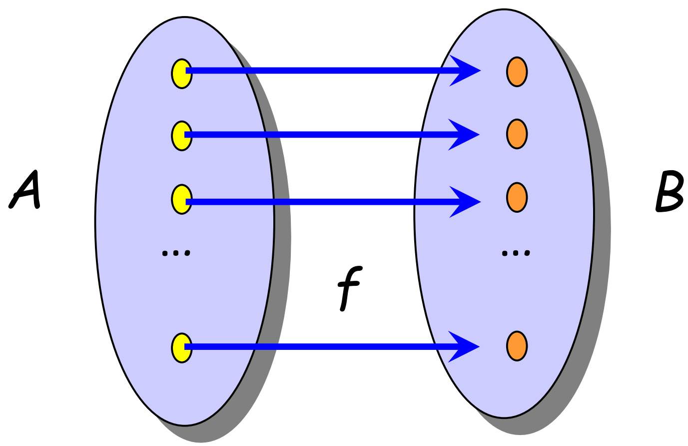
For the set of restricted 6-sequences,
there are 8 choices for r(p) and c(p),
there are 7 choices for r(k) and c(k),
there are 6 choices for r(b) and c(b).

Thus, total number of configurations $= (8 \times 7 \times 6)^2 = 112896$.

Counting Rule: Bijection

If f is a bijection from A to B,

then $|A| = |B|$



Steps:

1) Come up with B.
2) Come up with f.
3) Show f is a bijection.
4) Compute |B|.

Usually the first two steps are more difficult.

Now we see some more interesting examples.

Counting Doughnut Selections

There are five kinds of doughnuts.

How many different ways to select a dozen doughnuts?

00	(none)	000000	00	00
Chocolate	Lemon	Sugar	Glazed	Plain

A ::= all selections of a dozen doughnuts

Hint: define a bijection to some bit strings!

Counting Doughnut Selections

A ::= all selections of a dozen doughnuts

B::= all 16-bit binary strings with exactly four 1s.

Define a bijection f between A and B.

0011000000100100

00 1 1 000000 1 00 1 00

00	(none)	000000	00	00
Chocolate	Lemon	Sugar	Glazed	Plain

The 0s are used to represent the doughnuts, separated by four 1s to indicate different types of doughnuts.

Counting Doughnut Selections

A ::= doughnuts selections *B* ::= all 16-bit strings with four 1s.

Chocolate	Lemon	Sugar	Glazed	Plain		
12	0	0	0	0	$\longrightarrow$	0000000000001111
2	3	0	3	4	$\longrightarrow$	0010001100010000
2	0	6	2	2	$\longrightarrow$	0011000000100100
0	0	0	0	12	$\longrightarrow$	1111000000000000

Counting Doughnut Selections

c chocolate, l lemon, s sugar, g glazed, p plain maps to

$$0^c 1 0^l 1 0^s 1 0^g 1 0^p$$

A ::= all selections of a dozen doughnuts

B ::= all 16-bit binary strings with exactly four 1s.

This is a bijection because

- Two doughnut selections map to two different strings. (injection)

  (think of a doughnut selection as a sequence of 5 numbers (c,l,s,g,p),

  consider the first number that is different in the two sequences)

- Every string in B corresponds to a valid doughnut selection. (surjection)

Counting Doughnut Selections

c chocolate, l lemon, s sugar, g glazed, p plain maps to

$$0^c 1 0^l 1 0^s 1 0^g 1 0^p$$

A ::= all selections of a dozen doughnuts

B ::= all 16-bit binary strings with exactly four 1s.

$$|A| = |B| = \binom{16}{4}$$

Counting Loops

```
for i=1 to n do
        for j=1 to i do
                for k=1 to j do
                        printf("hello world\n");
```

How many "hello world"s will this program print?

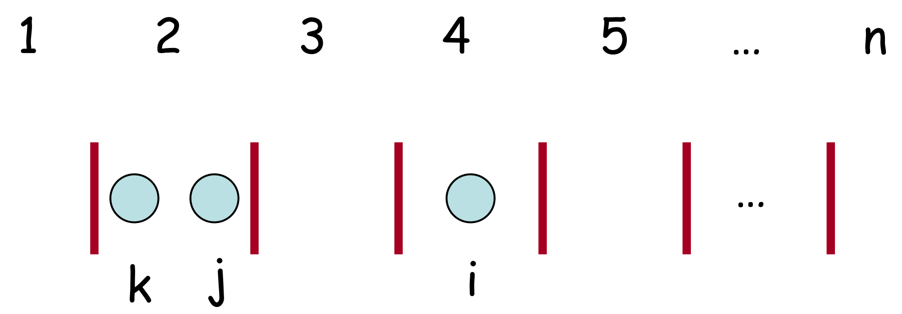
A ::= all (i,j,k)-triples

B ::= all strings with n-1 ones and 3 zeroes

Counting Loops

```
for i=1 to n do
        for j=1 to i do
                for k=1 to j do
                        printf("hello world\n");
```

There are n possible values for the i,j,k.



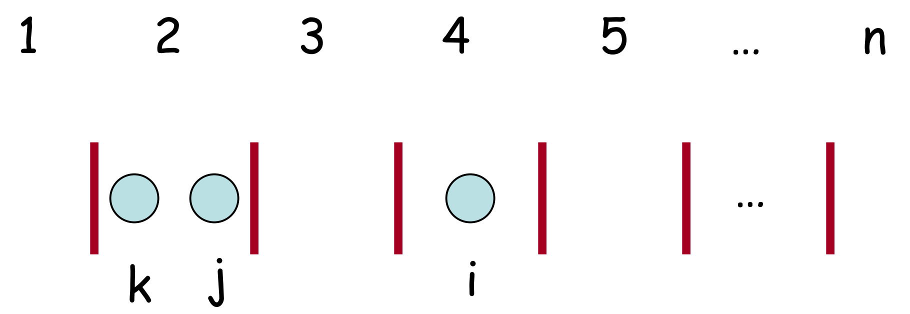
Imagine there are n-1 separators for the n values.

If i=4, j=2, k=2, then there are two zeroes in 2 and one zero in 4.

Counting Loops

```
for i=1 to n do
        for j=1 to i do
                for k=1 to j do
                        printf("hello world\n");
```

There are n possible values for the i,j,k.



The position of the first zero corresponds to the value of k.

The position of the second zero corresponds to the value of j.

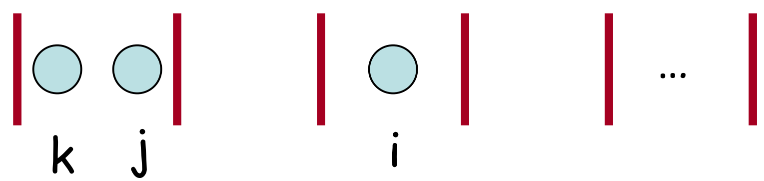
The position of the third zero corresponds to the value of i.

Counting Loops

```
for i=1 to n do
        for j=1 to i do
                for k=1 to j do
                        printf("hello world\n");
```

There are n possible values for the i,j,k.

1 2 3 4 5 … n



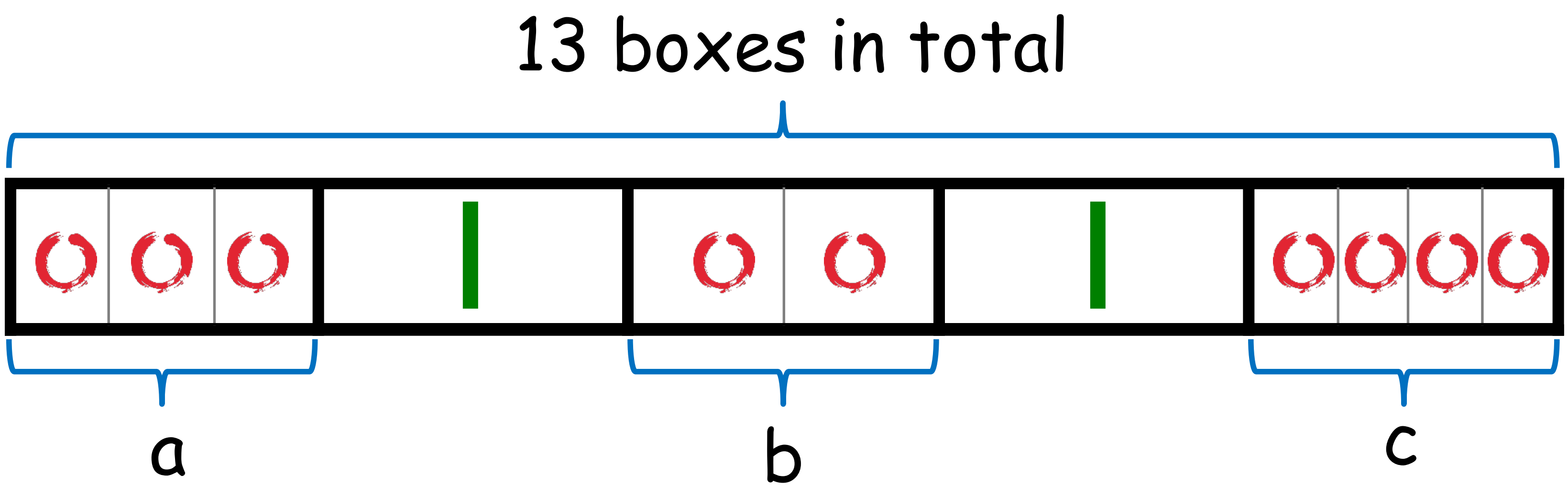
This is a bijection (verify this!) between the set of (i,j,k)-triples and the set of strings with n-1 ones and 3 zeros.

So, the program prints "hello world" exactly $\binom{n+2}{3}$ times.

Counting Non-Negative Integer Solutions

The universal set U = { (a,b,c) | a+b+c=11 }, let N = |U|.

Remember how to count N?



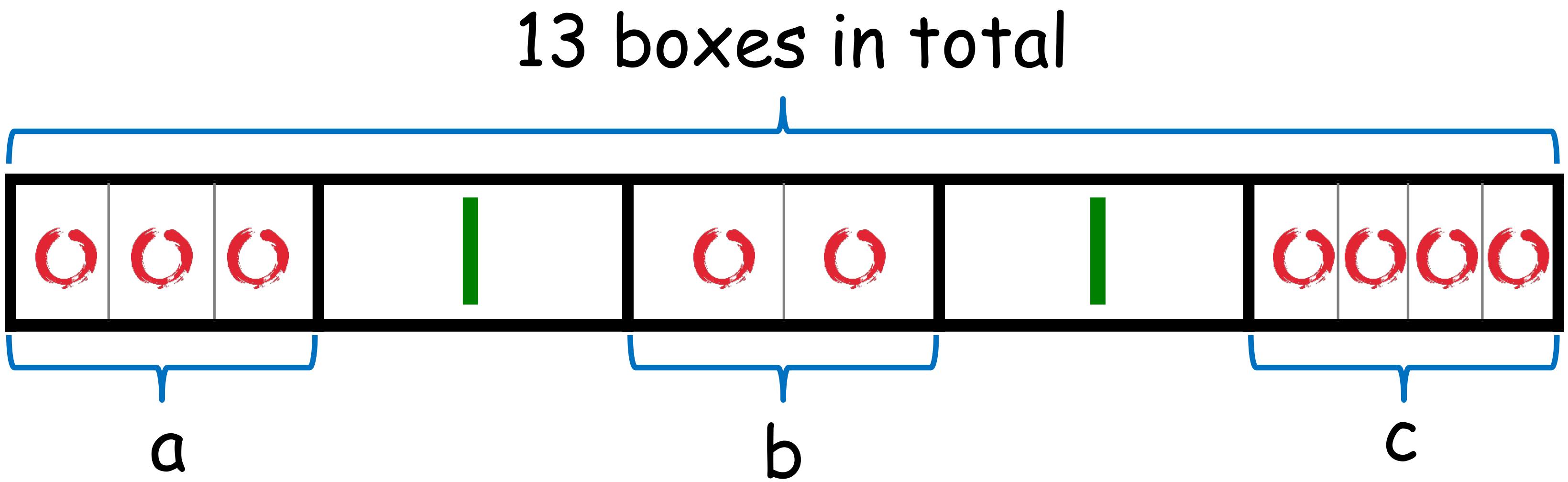
are placed in the boxes above, and separated by .

So N = $\binom{11+3-1}{11}$ = 78.

Counting Non-Negative Integer Solutions

The universal set U = { (a,b,c) | a+b+c=11 }, let N = |U|.

Remember how to count N?



A ::= all non-negative integer solutions for a+b+c=11

B ::= all 13-bit binary strings with exactly eleven 0s.

Counting Non-Negative Integer Solutions

How many integer solutions for $a+b+c = 11$ if $a,b,c \geq 1$?

Set $a=x+1$, $b=y+1$, $c=z+1$.

Consider the equation $x+y+z=8$ where $x,y,z >= 0$.

There is a bijection (**verify this!**) between the solutions for a,b,c and the solutions for x,y,z.

Therefore, we can apply the previous method to conclude that the answer is $\binom{10}{8}$.

Counting Non-Negative Integer Solutions

How many integer solutions for $a+b+c \leq 11$ if $a,b,c \geq 0$?

Consider the equation $a+b+c+d=11$ where $a,b,c,d >= 0$.

There is a bijection between the solutions for a,b,c and the solutions for a,b,c,d.

Therefore, we can apply the previous result to conclude

that the answer is $\binom{14}{11}$

because we need a 14-bit string with 11 zeroes.

Choosing Non-Adjacent Books

There are 20 books arranged in a row on a shelf.

How many ways to choose 6 of these books so that no adjacent books are selected?

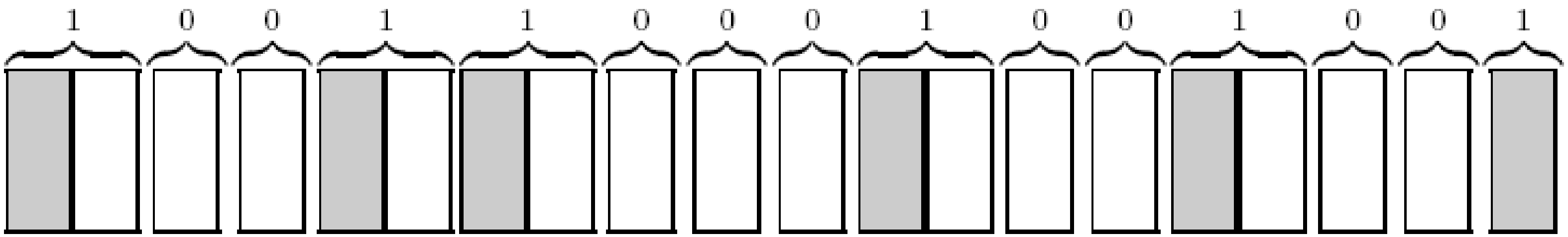
A ::= all selections of 6 non-adjacent books from 20 books

B ::= all 15-bit binary strings with exactly six **1**s.

Choosing Non-Adjacent Books

A ::= all selections of 6 non-adjacent books from 20 books

B ::= all 15-bit binary strings with exactly six **1**s.

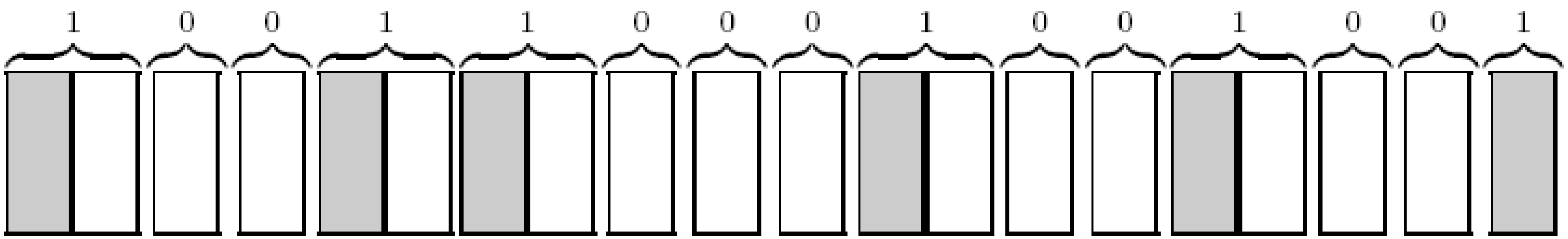


Map each **0** to a non-chosen book, each of the first five **1**s to a chosen book followed by a non-chosen book, and the last **1** to a chosen book.

This is a bijection from B to *A*, because:

- Different strings are mapped to different selections (injection)
- Every valid selection corresponds to a string in B (surjection)

Choosing Non-Adjacent Books



A ::= all selections of 6 non-adjacent books from 20 books

B ::= all 15-bit binary strings with exactly six 1s.

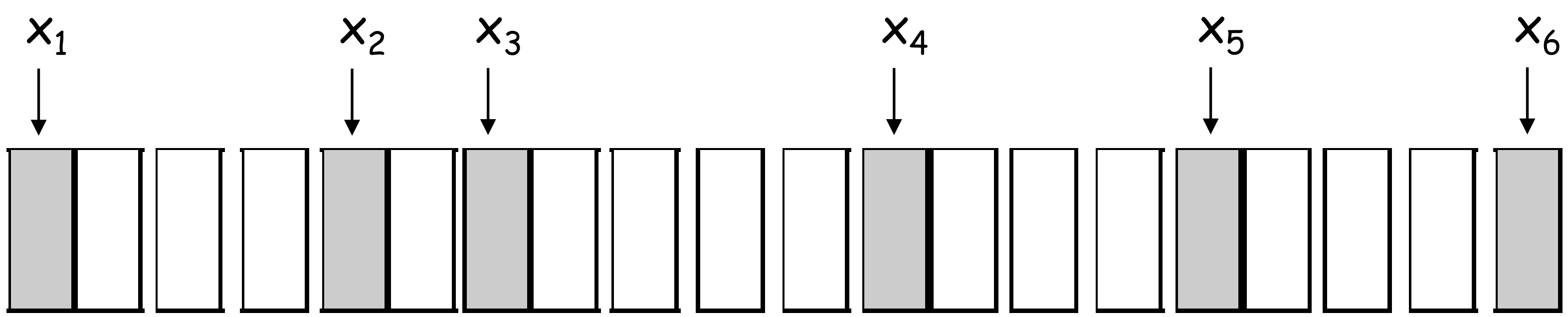
$$|A| = |B| = \binom{15}{6}$$

Choosing Non-Adjacent Books

A ::= all selections of 6 non-adjacent books from 20 books

B ::= the set of integer solutions for $x_1+x_2+x_3+x_4+x_5+x_6 \leq 20$ where $x_1 \geq 1$ and $x_2,x_3,x_4,x_5,x_6 \geq 2$.

Here, x_i represents picking the x_i-th book after the earlier chosen book.



e.g. this configuration corresponds to $x_1=1$, $x_2=4$, $x_3=2$, $x_4=5$, $x_5=4$, $x_6=4$.

It is not difficult to check that this is a bijection.

Choosing Non-Adjacent Books

A ::= all selections of 6 non-adjacent books from 20 books

B ::= the set of integer solutions for $x_1+x_2+x_3+x_4+x_5+x_6 \leq 20$
where $x_1 \geq 1$ and $x_2,x_3,x_4,x_5,x_6 \geq 2$.

From slide 34 we learned that $|B| = |C|$ for

C ::= the set of integer solutions for $x_1+x_2+x_3+x_4+x_5+x_6 \leq 9$
where $x_1,x_2,x_3,x_4,x_5,x_6 \geq 0$.

From slide 35 we learned that $|C| = |D|$ for

D ::= the set of integer solutions for $x_1+x_2+x_3+x_4+x_5+x_6+x_7 = 9$
where $x_1,x_2,x_3,x_4,x_5,x_6,x_7 \geq 0$.

From slide 32 we learned that $|D| = \binom{9+7-1}{9}$.

Exercises

How many non-negative integer solutions for $x_1+x_2+\ldots+x_m = n$?

How many integer solutions for $x_1+x_2+\ldots+x_m = n$
where $x_1 \geq a_1, x_2 \geq a_2, \ldots, x_m \geq a_m$?

How many integer solutions for $x_1+x_2+\ldots+x_m < n$
where $x_1 \geq a_1, x_2 \geq a_2, \ldots, x_m \geq a_m$?

Exercises

How many "hello world"s will this program print?

```
for i=1 to n do
        for j=1 to i-1 do
                for k=1 to j-1 do
                        printf("hello world\n");
```

How many non-negative integer solutions for $x_1+x_2 = n$

where $x_1 < a_1$ and $x_2 < a_2$?

(Hint 1: count the complement.)

(Hint 2: inclusion-exclusion.)

This Lecture

- Functions
- Bijection rule
- Division rule
- Catalan number

Division Rule

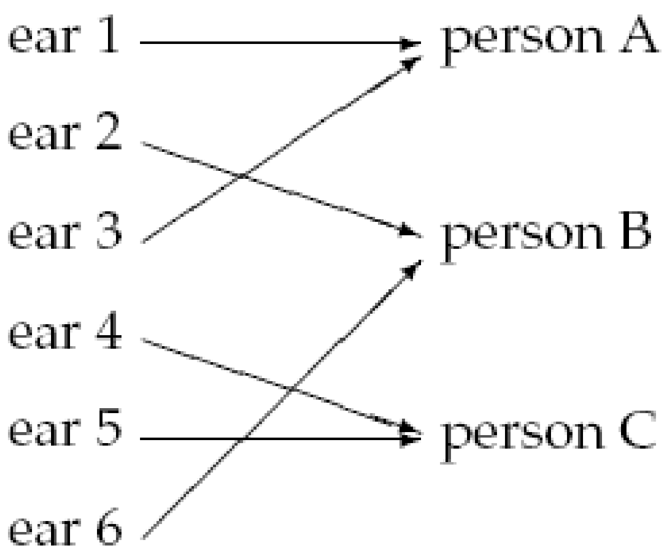
If a function from *A* to *B* is *k*-to-1,

meaning that each element in B is mapped by exactly k elements in *A*

then $|A| = k|B|$

(This generalizes the bijection rule.)

e.g.



This is a 2-to-1 function.

So #ears = 2 $\times$ #people.

Example: Handshaking Lemma

This is something we have encountered in graph theory.

Double Count!

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

A: the set of arcs.

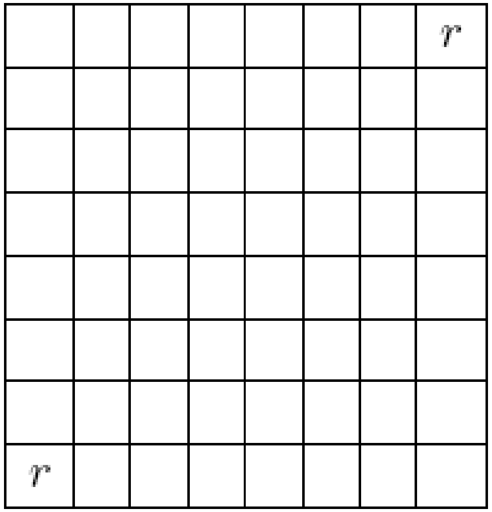
B: the set of edges.

The idea was to show that the mapping from A to B is 2-to-1,

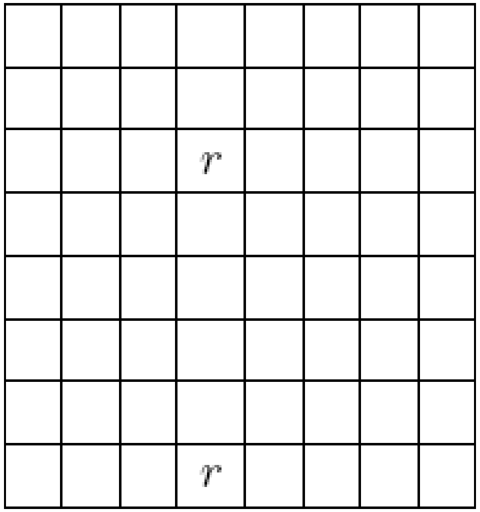
and thus conclude that $|A| = 2|B|$. Then we can compute $|A|$ by $|B|$.

Another Chess Problem

In how many different ways can you place two identical rooks on a chessboard so that they do not share a row or column?



valid



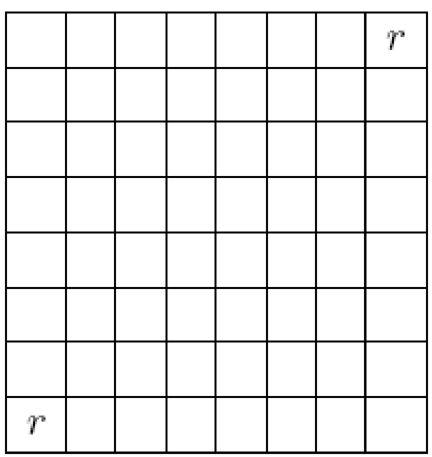
invalid

Another Chess Problem

We define a mapping between configurations
to sequences (r(1), c(1), r(2), c(2)),
where r(1) and r(2) are distinct rows,
and c(1) and c(2) are distinct columns.

A ::= all sequences (r(1),c(1),r(2),c(2)) with r(1) ≠ r(2) and c(1) ≠ c(2)

B ::= all valid rook configurations



(1,8,8,1) and (8,1,1,8) maps
to the same configuration.

This mapping is 2-to-1.

Another Chess Problem

A ::= all sequences (r(1),c(1),r(2),c(2)) with r(1) ≠ r(2) and c(1) ≠ c(2)

B ::= all valid rook configurations

This mapping is 2-to-1. ⇒ $$|A| = 2|B|$$

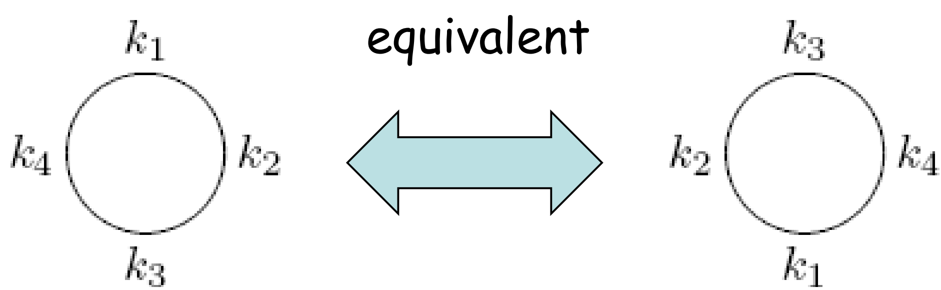
There are 8 choices of r(1) and c(1),
there are 7 choices of r(2) and c(2),
so |A| = 8x8x7x7 = 3136.

Thus, total number of configurations
|B| = |A|/2 = 3136/2 = 1568.

Round Table

How many ways can we seat n different people at a round table?

Two seatings are considered *equivalent* if one can be obtained from the other by a rotation.

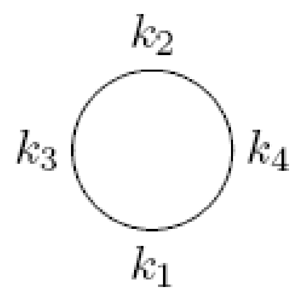


Round Table

A ::= all the permutations of the n people

B ::= all possible seating arrangements at the round table

$$(k_2, k_4, k_1, k_3) \longrightarrow$$



Map each permutation in set *A* to a circular seating arrangement in set B by following the ordering in the permutation.

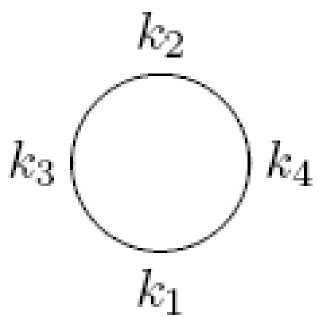
Round Table

A ::= all the permutations of the n people

B ::= all possible seating arrangements at the round table

$$(k_2, k_4, k_1, k_3)$$
$$(k_4, k_1, k_3, k_2)$$
$$(k_1, k_3, k_2, k_4)$$
$$(k_3, k_2, k_4, k_1)$$

$\longrightarrow$



This mapping is n-to-1.

$$|A| = n|B|$$

Thus, total number of seating arrangements is
|B| = |A|/n = n!/n = (n-1)!

Counting Subsets

Now we can compute $\binom{n}{k}$ more formally.

How many 4-subsets in {1,2,…,13}?

Let A ::= permutations of {1,2,…,13}

B ::= 4-subsets in {1,2,…,13}

map $a_1\, a_2\, a_3\, a_4\, a_5 \ldots\, a_{12}\, a_{13}$ to $\{a_1, a_2, a_3, a_4\}$

(that is, take the first k elements from the permutation)

How many permutations are mapped to the same subset??

Counting Subsets

map $a_1\, a_2\, a_3\, a_4\, a_5 \ldots a_{12}\, a_{13}$ to $\{a_1, a_2, a_3, a_4\}$

$a_2\, a_4\, a_3\, a_1\, a_5 \ldots a_{12}\, a_{13}$ is also mapped to $\{a_1, a_2, a_3, a_4\}$

So is $\underbrace{a_2\, a_4\, a_3\, a_1}_{4!}\, \underbrace{a_{13}\, a_{12} \ldots a_5}_{9!}$

Any ordering of the first four elements (4! of them),
and any ordering of the last nine elements (9! of them)
will give the same subset.

So this mapping is $4! \times 9!$-to-1 $\Rightarrow$ $$|A| = 4!9!|B|$$

Counting Subsets

Let A ::= permutations of {1,2,…,13}

B ::= 4-subsets in {1,2,…,13}

$$|A| = 4!9!|B|$$

$$13! = |A| = 4!9!|B|$$

So number of 4-subsets is $\binom{13}{4} ::= \frac{13!}{4!9!}$

Number of m-subsets in an n element set is

$$\binom{n}{m} ::= \frac{n!}{m!(n-m)!}$$

MISSISSIPPI

How many different spellings by rearranging letters in "MISSISSIPPI"?

Let A ::= {permutations of the 11 letters}

B ::= {spellings by rearranging letters in "MISSISSIPPI"}

How many permutations are mapped to the same spelling?

MISSISSIPPI →

- 4! permutations of the "**S**"s gives the same spelling
- 4! permutations of the "**I**"s gives the same spelling
- 2! permutations of the "**P**"s gives the same spelling

The mapping is 4!4!2!-to-1, and so there are 11!/4!4!2! different words.

Example: 20-Mile Walk

A robot is planning a 20-mile walk, consisting of 5 of one-mile northward walks, 5 of one-mile southward walks, 5 of one-mile eastward walks, and 5 of one-mile westward walks.

How many different walks are possible?

There is a bijection between such walks and words with 5 N's, 5 E's, 5 S's, and 5 W's.

The number of such words is equal to the number of rearrangements:

$$\frac{20!}{5!5!5!5!}$$

Exercises

What is the coefficient of $x^7y^9z^5$ in $(x+y+z)^{21}$?

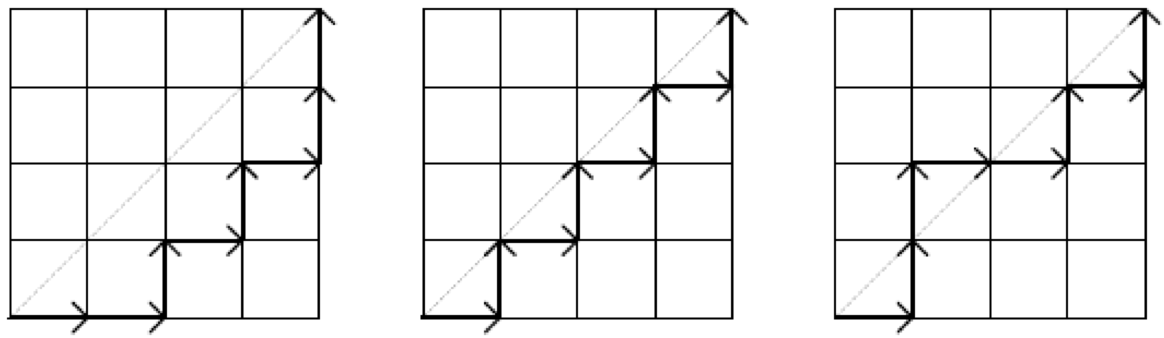
There are 12 people. How many ways to divide them into 3 teams, each team with 4 people?

This Lecture

- Functions
- Bijection rule
- Division rule
- Catalan number

Monotone Path

A ***monotone path*** from (0,0) to (n,n) is a path consisting of "right" moves (x-coordinate +1) and "up" moves (y-coordinate +1), starting at (0,0) and ending at (n,n).



How many possible monotone paths from (0,0) to (n,n)?

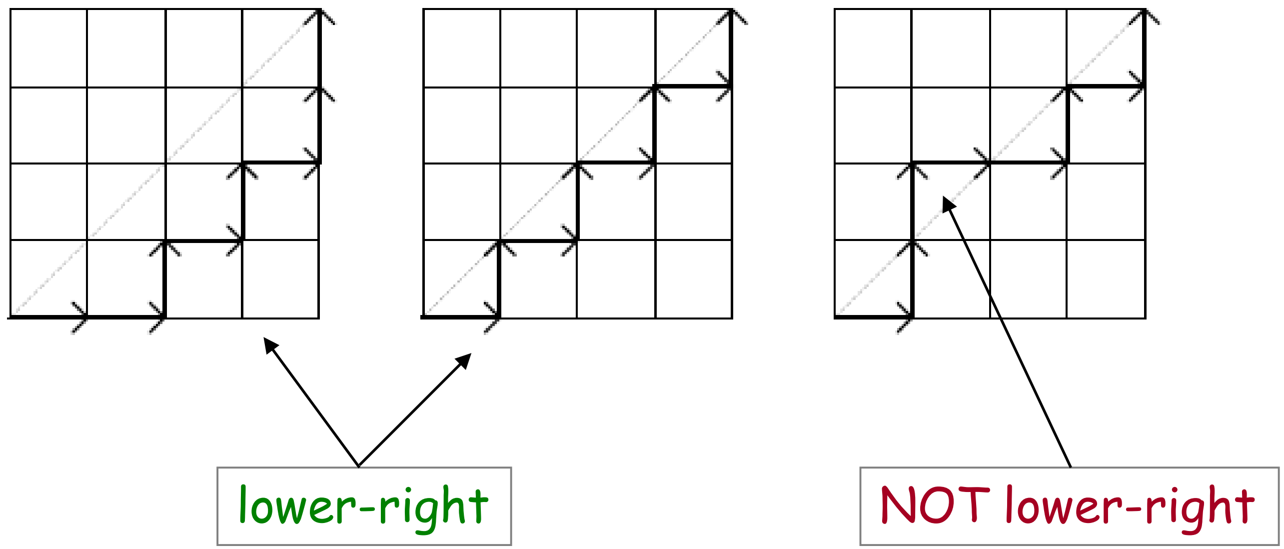
We can map a "right" move to an "x" and a "up" move to a "y".

There is a bijection between monotone paths and words with n x's and n y's.

And so the answer is just $\binom{2n}{n}$.

Monotone Path

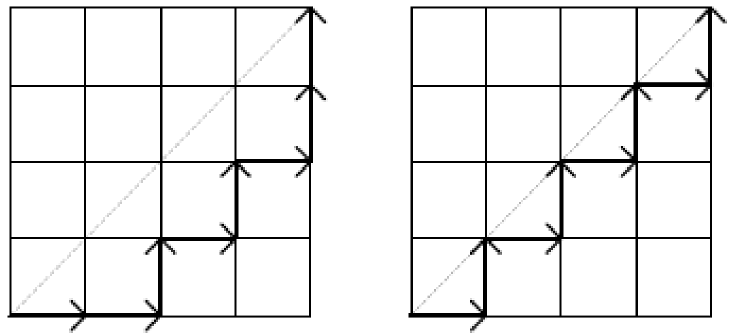
A monotone path is called ***lower-right*** if any point (x_i,y_i) on the path has $x_i \geq y_i$.



How many possible *lower-right* monotone paths from (0,0) to (n,n)?

Monotone Path

How many possible *lower-right* monotone paths from (0,0) to (n,n)?



We can still map a "right" move to an "x" and a "up" move to a "y".

There is a bijection between lower-right monotone paths (*A*) and words with n x's and n y's (B), with the additional constraint that

no initial segment of the string has more y's than x's.

The set B is called the set of **Dyck words**.

This is a bijection, but both sets look difficult to count.

Parenthesis

How many valid ways to add n pairs of parentheses?

E.g. There are 5 valid ways to add 3 pairs of parentheses.

((()))   (()())   (())()   ()(())   ()()()

Let r_n be the number of ways to add n pairs of parentheses.

A pairing is ***valid*** if the number of open parentheses is never less than number of close parentheses from the left.

Remember this notorious Catalan number r_n?

Mountain Ranges

How many "mountain ranges" can you form with n upstrokes and n downstrokes above a horizontal line?

```
                                          /\
             /\      /\        /\/\      /  \
/\/\/\,   /\/  \,   /  \/\,   /    \,   /    \
```

We do not know how to solve these four problems yet,
(A) lower-right monotone paths, (B) Dyck words,
(C) valid parentheses, (D) mountain ranges.

But we can show that all these four problems have the same answer, by showing that there are bijections between these sets.

Parenthesis and Monotone Paths

A pairing is ***valid*** if the number of open parentheses is never less than number of close parentheses from the left.

(()()())   ()()()()

⇕   ⇕

xxyxyxyy   xyxyxyxy

We can map a "(" to an "x" and a ")" move to a "y".

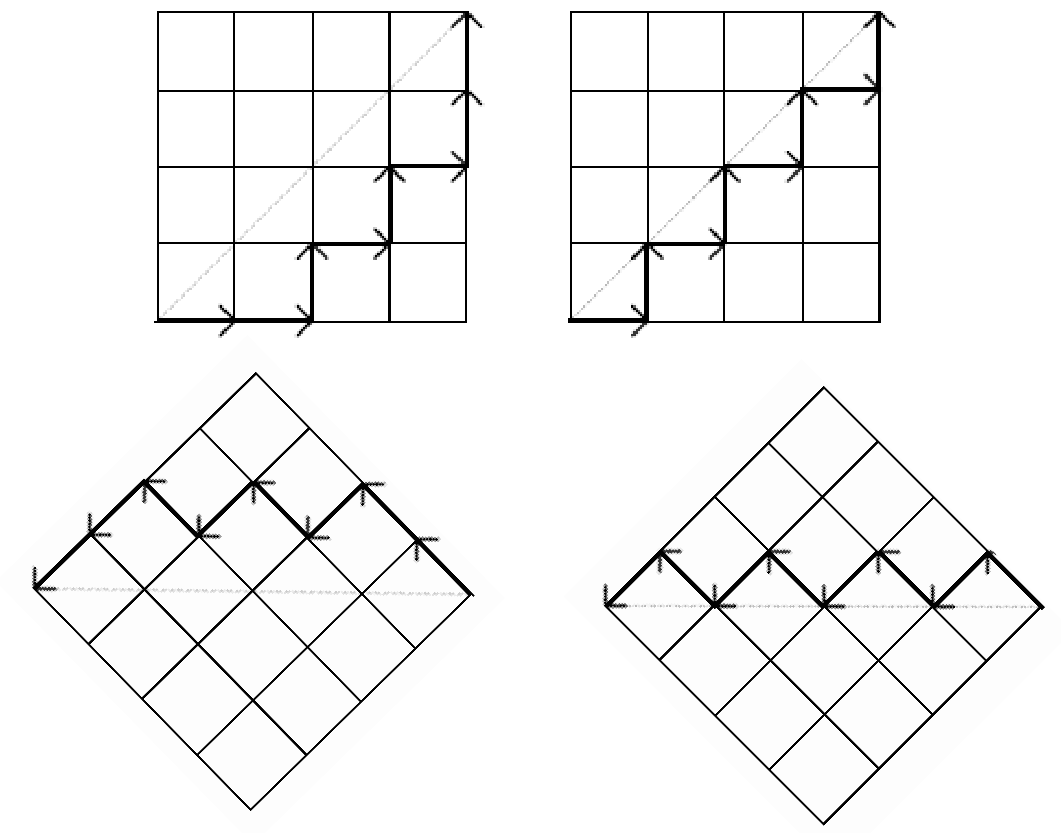
This is a bijection between (*C*) valid pairings and (B) Dyck words.

In slide 61, we have seen that there is a bijection between

(*A*) lower-right monotone paths and (B) Dyck words.

So we can find a bijection between (*A*) and (*C*).

Monotone Paths and Mountain Ranges



By "rotating" the images, we see that a path not crossing the diagonal is just the same as a mountain not crossing the horizontal line.

So there is a bijection between (*A*) and (D).

Catalan Number

Now we know that these four sets are all of equal size,

which is our lovely Catalan number

$$r_n = \frac{1}{n+1}\binom{2n}{n}$$

Can we find a slick proof for this formula?

One of the proofs is by the bijection rule.

Proof Plan

The number of lower right monotone paths from (0,0) to (n,n) is $\frac{1}{n+1}\binom{2n}{n}$.

Our plan is to count the complement.

In slide 59, we know that the number of monotone paths is $\binom{2n}{n}$

We will show that the number of **non-**lower-right monotone paths is $\binom{2n}{n+1}$

This would imply that the number of lower-right monotone paths is

$$\binom{2n}{n}-\binom{2n}{n+1}=\frac{(2n)!}{n!n!}-\frac{(2n)!}{(n+1)!(n-1)!}=\frac{(2n)!}{(n+1)!n!}=\frac{1}{n+1}\binom{2n}{n}$$

Proof

The number of non-lower-right monotone paths is $\binom{2n}{n+1}$.

The idea is to define a bijection between

(*A*) The set of non-lower-right monotone paths from (0,0) to (n,n)

(B) The set of monotone paths from (0,0) to (n-1,n+1)

Clearly, |B| = $\binom{2n}{n+1}$,

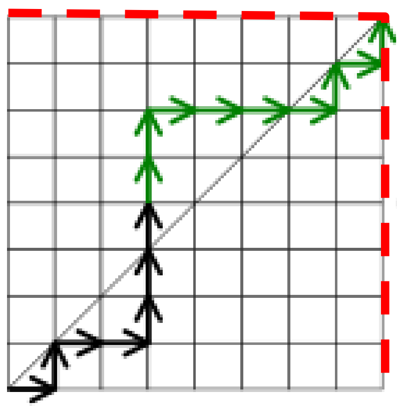
as it is equal to the set of strings with 2n characters,

where n-1 of them are R ("right") and n+1 of them are U ("up").

So, it remains to define a bijection between (*A*) and (B).

Injection

Every path in (A) must "cross" the diagonal at least once.
We look at the first "crossing", and then "flip the path".



(picture from wiki)

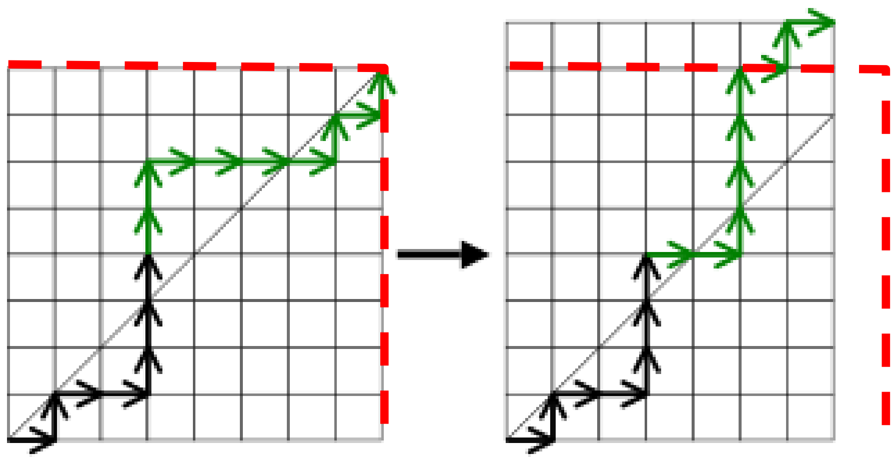
In the original path, before the flipping point, we have #U = #R + 1.
So, in the original path, after the flipping point, we have #R = #U + 1.

In the new path, since we flip the latter part, we have #U = #R + 2,
and thus it is a path from (0,0) to (n-1,n+1).

Moreover, this map is injective. (Why?)

Surjection

Every path in (A) must "cross" the diagonal at least once.
We look at the first "crossing", and then "flip the path".



(picture from wiki)

In order to check the map is surjective we can check the map is invertible.

Given a monotone path from (0,0) to (n-1,n+1), it must cross the diagonal. (Why?)
Look at the first such point and flip it, we get back a non-lower-right monotone path from (0,0) to (n,n), which is the preimage of the map.

Quick Summary

Counting by mapping is a very useful technique.

It is also powerful for solving many complicated problems.

The basic examples usually map a set onto a properly defined binary strings.

Then we see how to generalize this approach by considering k-to-1 functions.

Finally we see the mappings between more complicated sets.

Anyway, the idea of defining a mapping (reduction) between two problems

is probably the most important idea in computer science.

So, make sure you are familiar with it.