

1. Mapping rule

2. Division rule

## COSC 290 Discrete Structures

### Lecture 28: Division rule and pigeonhole principle

Prof. Michael Hay

Wednesday, Apr. 11, 2018

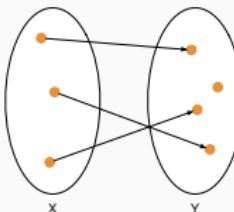
Colgate University

1

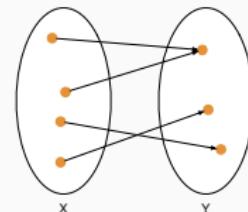
## Mapping rule

### One-to-one

One-to-one: for every  $y \in Y$ , there is at most one  $x \in X$  such that  $f(x) = y$ .



$f$  is one-to-one  $\implies |X| \leq |Y|$

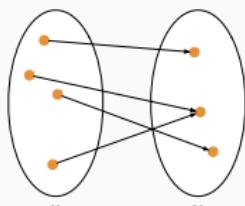


$f$  is not one-to-one and  $|X| \not\leq |Y|$

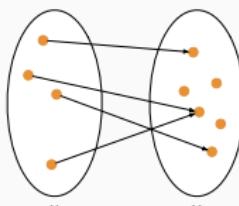
2

## Onto

Onto: for every  $y \in Y$ , there is **at least one**  $x \in X$  such that  $f(x) = y$ .



$f$  is **onto**  $\implies |X| \geq |Y|$

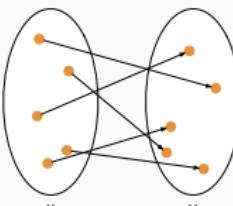


$f$  is **not onto** and  $|X| \geq |Y|$

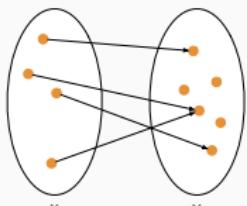
3

## Bijective

Bijective: for every  $y \in Y$ , there is **exactly one**  $x \in X$  such that  $f(x) = y$ .



$f$  is **bijective**  $\implies |X| = |Y|$



$f$  is **not bijective** and  $|X| \neq |Y|$

4

## Example: applying the mapping rule

Let  $B_n \subset \{[], []\}^n$  be a set of strings consisting of balanced brackets.

Examples:

$$B_4 = \{[], [], [[], []], [][], [][[]], [][[]], [][[]][], [][[]][[]]\}$$

$$B_5 = \{[], [], [[], []], [][], [][[]], [][[]], [][[]][], [][[]][[]], [][[]][[]][], [][[]][[]][[]]\}$$

**Claim:**  $2^{n/4} \leq |B_n| \leq 2^n$ .

We will prove this using the mapping rule (repeated below):

Let  $f : X \rightarrow Y$ .

- $|X| \geq |Y|$  if and only if  $f$  is onto.
- $|X| \leq |Y|$  if and only if  $f$  is one-to-one.
- $|X| = |Y|$  if and only if  $f$  is bijective (onto and one-to-one).

5

## Poll: only so many friends

Let  $f : \text{Friends} \rightarrow \{-1, 102, 201, 290, 301, 302, 304, 460, 480\}$  be a function that maps each person in *Friends* to the highest numbered required COSC course they have taken (-1 if none).

When applied to your friends, suppose the maximum value is 301. Therefore you have at most 5 friends. **What's wrong with this claim?**

- A)  $f$  is not necessarily bijective
- B)  $f$  is not necessarily onto
- C)  $f$  is not necessarily one-to-one
- D) The logic is correct, but claim is false because one of my friends took 304!

Mapping rule: Let  $f : X \rightarrow Y$ .

- $|X| \geq |Y|$  if and only if  $f$  is onto.
- $|X| \leq |Y|$  if and only if  $f$  is one-to-one.
- $|X| = |Y|$  if and only if  $f$  is bijective (onto and one-to-one).

6

## Division rule

### Example: division rule

Spatial indexing (next two slides). To efficiently support map services (Google maps, Yelp “nearby”), data stored in spatial indexes.

A **rectangular range query** specifies a type of item (e.g., taqueria) and a *rectangle* and returns all such items that overlap with the given rectangle.

Assume city is divided into  $n \times n$  discrete points. Rectangle must include at least one point. **How many rectangle queries are possible?**

#### Definition (*k*-to-1 functions)

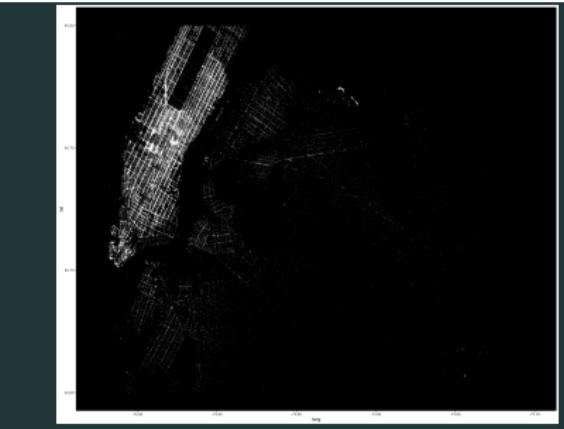
Let  $f : X \rightarrow Y$ . We say that  $f$  is ***k*-to-1** if for all  $y \in Y$ , there are  $k$  distinct elements of  $X$  such that  $f(x) = y$ . In other words,

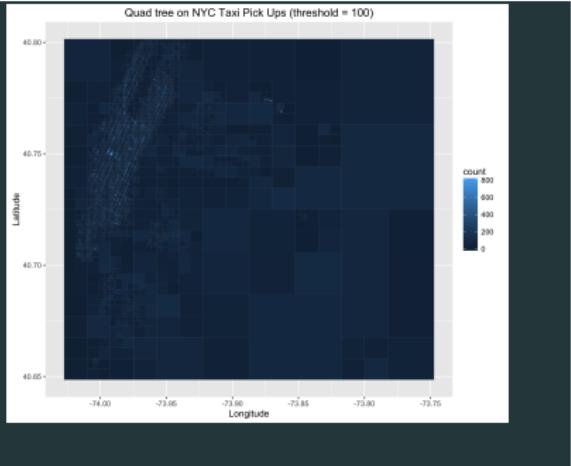
$$\forall y \in Y : |\{x \in X : f(x) = y\}| = k$$

#### Theorem (Division rule)

Let  $f : X \rightarrow Y$ . If  $f$  is *k*-to-1, then  $|X| = k \cdot |Y|$ .

(Draw picture on board)





## Counting rectangular range queries

A *rectangle query* is a 4-tuple  $\langle x_1, y_1, x_2, y_2 \rangle$  where  $x_1 \leq x_2$  and  $y_1 \leq y_2$ .

Cases:

1.  $x_1 < x_2$  and  $y_1 < y_2$
2.  $x_1 = x_2$  and  $y_1 < y_2$
3.  $x_1 < x_2$  and  $y_1 = y_2$
4.  $x_1 = x_2$  and  $y_1 = y_2$

These cases are disjoint and cover all possibilities, so we can count each one separately and apply the **sum rule** to get the total number of rectangle queries.

## Example: division rule

The **rectangular range query** counting problem.

Assume city is divided into  $n \times n$  discrete points,  $\{0, \dots, n-1\}^2$ .

A *rectangle query* is a 4-tuple  $\langle x_1, y_1, x_2, y_2 \rangle$  where  $x_1 \leq x_2$  and  $y_1 \leq y_2$ .

(Think of  $\langle x_1, y_1 \rangle$  as defining lower left corner of rectangle and  $\langle x_2, y_2 \rangle$  as defining upper right corner.)

The answer to a query is the total number of items located at any point  $\langle x, y \rangle$  such that  $x_1 \leq x \leq x_2$  and  $y_1 \leq y \leq y_2$ .

**How many rectangle queries are possible?**

9

### Case 1: $x_1 < x_2$ and $y_1 < y_2$

How many ways to choose two distinct  $x$  values? Let  $N := \{0, \dots, n-1\}$ .

Let  $S := \{(x_a, x_b) \in N \times N : x_a \neq x_b\}$  What is  $|S|$ ?

$$|S| = n \cdot (n-1)$$

But in rectangle query,  $x_1$  must be less than  $x_2$ . So for each pair in  $S$ , apply this function to swap out-of-order pairs:

$$f(\langle x_a, x_b \rangle) := \begin{cases} \langle x_a, x_b \rangle & \text{when } x_a < x_b \\ \langle x_b, x_a \rangle & \text{when } x_a > x_b \end{cases}$$

Let  $T := \{f(\langle x_a, x_b \rangle) : \langle x_a, x_b \rangle \in S\}$ . What is  $|T|$ ?

This function **f** is **2-to-1**, so by division rule,  $|T| = |S|/2 = n \cdot (n-1)/2$

Apply same reasoning to  $\langle y_1, y_2 \rangle$ . By product rule, total 4-tuples  $\langle x_1, x_2, y_1, y_2 \rangle$  is  $(n \cdot (n-1)/2)^2$

## Counting rectangular range queries continued...

A rectangle query is a 4-tuple  $(x_1, y_1, x_2, y_2)$  where  $x_1 \leq x_2$  and  $y_1 \leq y_2$ .

Cases:

1.  $x_1 < x_2$  and  $y_1 < y_2$ :  $(n \cdot (n - 1)/2)^2$  tuples
2.  $x_1 = x_2$  and  $y_1 < y_2$ :  $n^2 \cdot (n - 1)/2$  tuples
3.  $x_1 < x_2$  and  $y_1 = y_2$ :  $n^2 \cdot (n - 1)/2$  tuples
4.  $x_1 = x_2$  and  $y_1 = y_2$ :  $n^2$  tuples

Other cases are left as exercises... but answers are shown above.

## Exercise: Knights at a round table

How many ways can you *arrange*  $n$  knights at a round table?

A *seating* defines who sits where. Two seatings are considered same *arrangement* if each knight has the same knight on its left and right in both seatings.

Example: here are two distinct *seatings*, but they represent the same *arrangement*.



Images taken and problem adapted from Lehman et al. Mathematics for Computer Science, 2010.