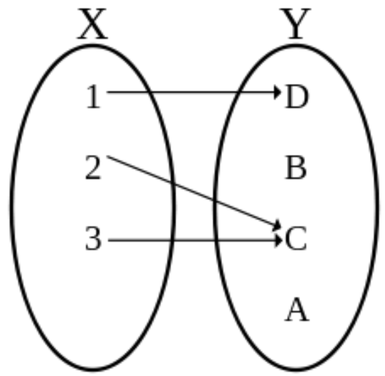

Counting
Sheep
Sets
Goal: Avoid counting!

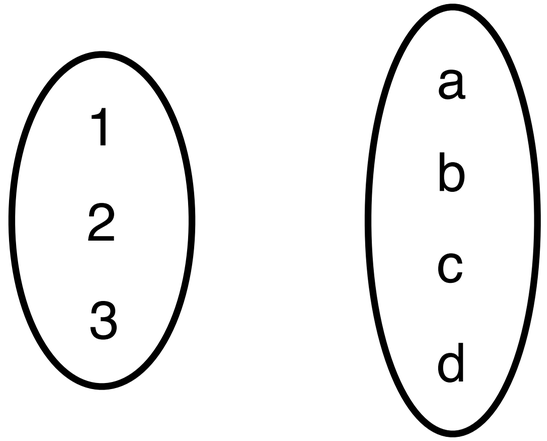
The Bijection Method

Functions

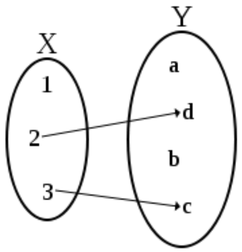
A function f from A to B, denoted f : A → B, associates with every a∈A an image f(a)∈B

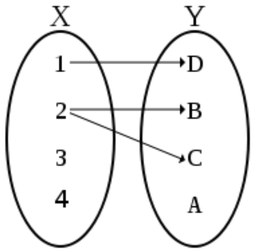


f : {1, 2, 3} → {a, b, c, d}



f(1)=b, f(2)=a, f(3)=d





Two A-elements can share image

What's not

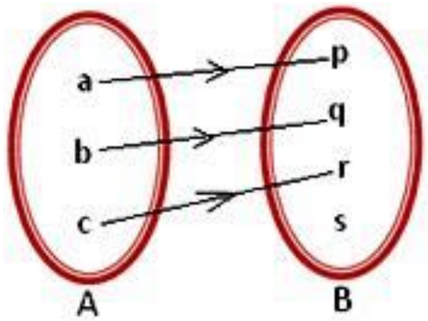
What's fun

1-1

"one to one"

f: A → B is 1-1, or injective, if different elements have different images

$\forall\ a, a' \in A$ $a \neq a' \rightarrow f(a) \neq f(a')$ $f(a) = f(a') \rightarrow a = a'$



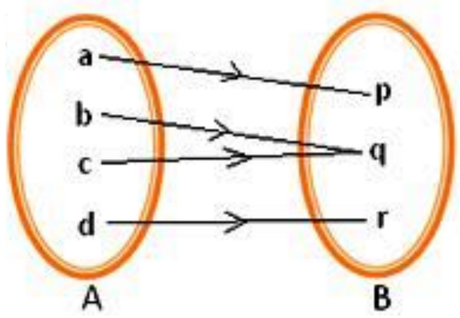
$f(a) \neq f(b)$
$f(a) \neq f(c)$
$f(b) \neq f(c)$

1-1

✅

✅

f: A → B is not 1-1 if $\exists\ a \neq a' \in A$, $f(a) = f(a')$



$f(b)=f(c)$

not 1-1

Set Size

The number of elements in a set S is called its size, or cardinality, and denoted |S| or # S

Notætion

Set of size n: n-set

Bits $|\{0,1\}| = 2$

Coin $|\{\text{heads, tails}\}| = 2$

Die $|\{1,2,3,4,5,6\}| = 6$

Digits $|\{0,1,\ldots,9\}| = 10$

Letters $|\{a,\ldots,z\}| = 26$

Empty set $|\varnothing| = 0$

Integers $|\mathbb{Z}| = |\mathbb{N}| = |\mathbb{P}| = \infty$

Countably infinite $\aleph_0$

Reals $|\mathbb{R}| = \infty$

Uncountably infinite $\aleph$

Integer Intervals

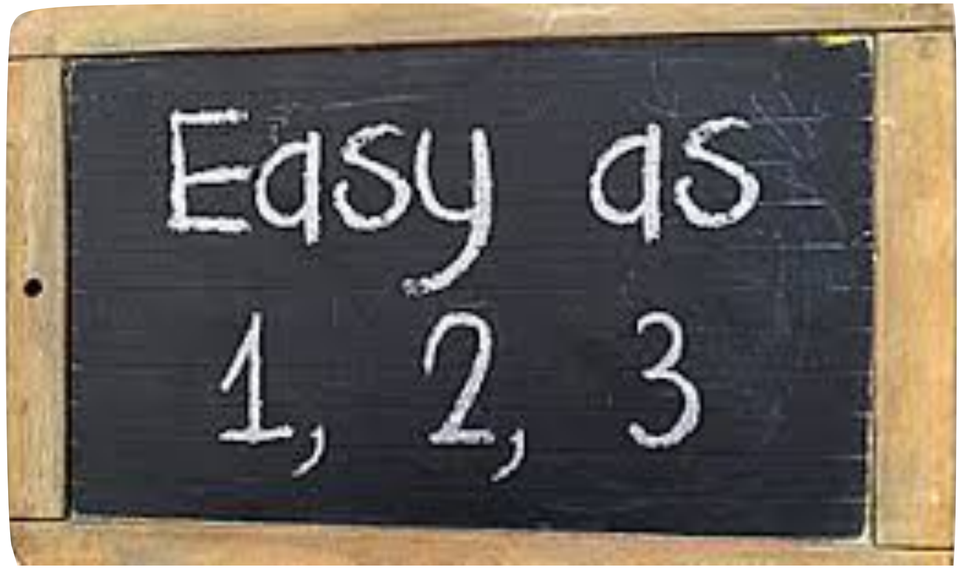
$m \le n$ $\{m,\ldots,n\}$ = {integers from m to n, inclusive} $\{3,\ldots,5\} = \{3,4,5\}$

$|\{m,\ldots,n\}| = n-m+1$

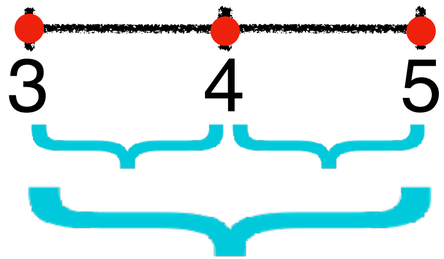
+1? First Try small #'s

$|\{5,\ldots,5\}| = |\{5\}| = 1 = 5 - 5 + 1$

$|\{1,\ldots,3\}| = |\{1,2,3\}| = 3 = 3 - 1 + 1$



$\{3,4,5\}$



points = #intervals+1= (5-3)+1 = 3

5-3 = 2 = length = #intervals

Integer Multiples

$_d(n] = \{\ 1 \le i \le n : d \mid i\ \}$

$(n] = [n] = \{\ 1, \ldots, n\ \}$

$_3(8] = \{\ 3, 6\ \} = \{1 \cdot 3, 2 \cdot 3\}$

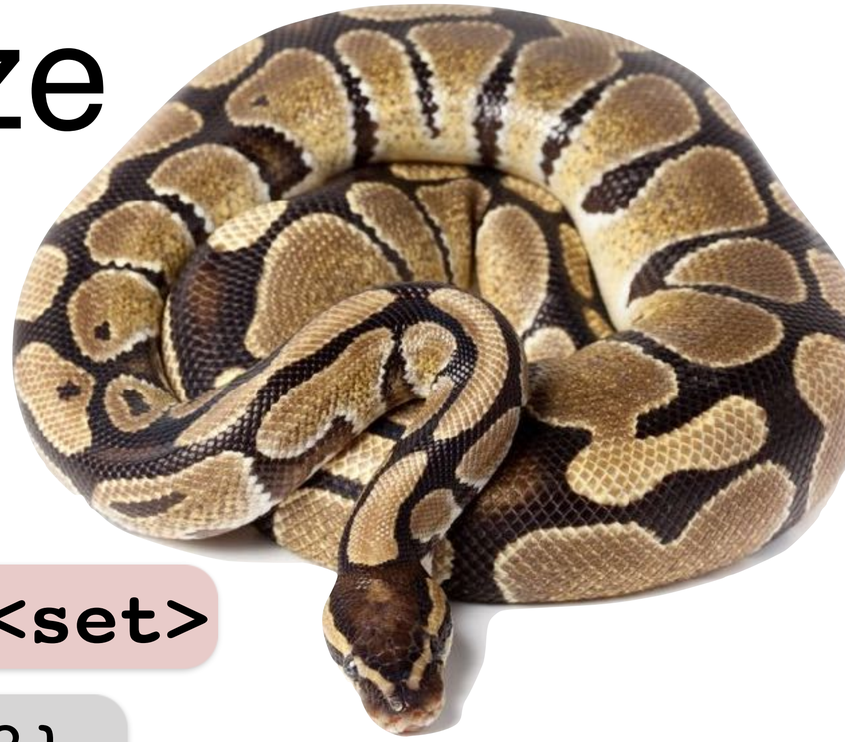
$_3(9] = \{\ 3, 6, 9\ \} = \{1 \cdot 3, 2 \cdot 3, 3 \cdot 3\}$

$| \ _d(n] \ | = \lfloor\ n/d\ \rfloor$

$| \ _3(8] \ | = \lfloor\ 8/3\ \rfloor = 2$

$| \ _3(9] \ | = \lfloor\ 9/3\ \rfloor = 3$

Set Size



Size `len`

```
print(len({-1, 1}))
2
```

Sum `sum`

```
print(sum({-1, 1}))
0
```

Minimum `min`

```
print(min({-1, 1}))
-1
```

Maximum `max`

```
print(max({-1, 1}))
1
```

Loops

`for <var> in <set>`

```
A = {1, 2, 3}
print(len(A))
3
num=0
for i in A:
    num += 1
print(num)
3
```