Graph Analytics

Jorge Finke
finke@ieee.org

Course overview

- Combinatorics (weeks 1-3)
  - Counting techniques (rules)
  - Tuples and permutations
  - Combinations
  - Binomial Theorem
  - Combinations with repetitions
  - Various problems in combinatorics

Implications of counting rules

- Discrete probability (week 4)
  - Random experiments
  - The notion of an event
  - Calculating probabilities
  - Outcomes with non-equal probabilities

Use of permutations and combinations to compute probabilities

Course overview

- The notion of graphs (week 5)
  - Graph database (Neo4j and Cypher)
  - Graph analysis in Python (Networkx)
- Path analytics (week 6)
  - How to move across nodes and edges?
- Connectivity analytics (week 7)
  - How are edges organized and structured?
- Community analytics (week 8)
  - Are there closely interacting nodes?
- Centrality analytics (week 9)
  - What are the most significant nodes?

Capture big data as graphs

Identify kinds questions that can be explored via graph analytics (probabilities)

Course overview

- Network formation models (weeks 10-12)
  - Small-world models
  - Scale-free models

Capture broad empirical patterns through models

- Machine learning on graphs (weeks 13-15)
  - Link prediction
  - Nodes type estimation

Core ideas of learning using graph data

No covered in this class

- Graph programming models (independent of programming language)
    - How do concurrently operating processes exchange data?
    - Share memory vs. pass messages (delays?)
    - What becomes parallel? What is done un parallel?
    - Decompose large task vs. fragment data?
- Bulk Synchronous Parallel (BSP)
    - Pregel (BSP data flow paradigm by Google)
    - GraphLab (by Carnegie Mellon University)
- Graph computing platforms
    - Giraph (BSP implementation on Hadoop)
    - GraphX (Spark's graph-processing API) - Scala
- Graph visualization

Combinatorics

Today

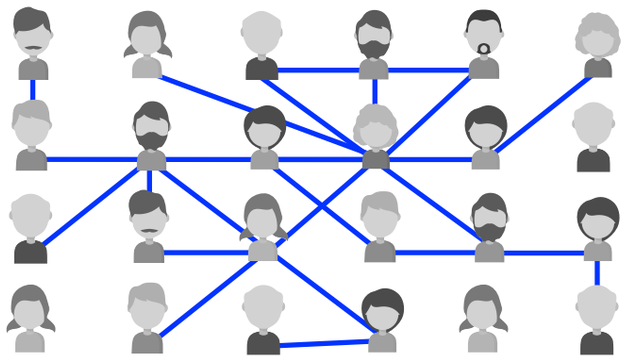
- Why counting?
- Rule of sum
- Convenient language: sets
- Generalizing rule of sum
- Recursive counting: number of paths
- Rule of product

Why counting?

- Tell how many objects/elements are there
- Without actually counting them one by one
- Application:
    - Counting number of steps in an algorithms
    - Estimating size of data
    - Computing probabilities

Examples

How many people are there?



4 039 nodes, 88 234 edges

What is the probability that there
is an edge between any two users (nodes)?

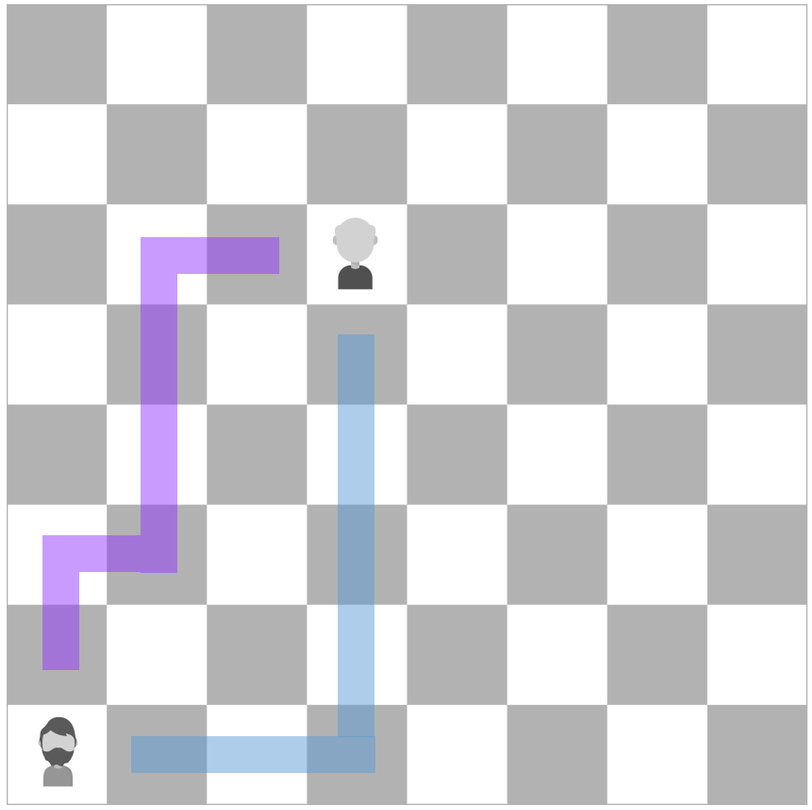
3 digits, 3 letters, city identifier

Are there enough license plates for everyone?

Rule of sum

Rule of sum

If there are k elements of a first type and n elements of a second type, then there are $n+k$ elements of the one of two types



Move of type 1: move right

Move of type 2: move up

In total, we always need 3 + 5 = 8 moves

Rule of sum

Rule of sum

If there are k elements of a first type and n elements of a second type, then there are $n+k$ elements of the one of two types

videos > 10min

videos < 10min

7 + 5 = 12 videos in total

videos > 4min

music videos

Are there 12 videos that are either longer than 4 minutes or that are music videos?

Not necessary, categories may overlap!

Rule of sum

Rule of sum

If there are k elements of a first type and n elements of a second type, then there are $n+k$ elements of the one of two types

No element should be of both types!

Exercise

How many times will the phrase “Hello, World!” be printed?

```
for _ in range(7):
    print('Hello, World!')
for _ in range(6):
    print('Hello, World!')
```

How many times will the phrase “Hello, World!” be printed?

```
for _ in range(8):
    print('Hello, World!')
for _ in range(4):
    print('Hello, World!')
for _ in range(7):
    print('Hello, World!')
```

Exercise

How many numbers from 1 to 20, inclusive, are divisible by 2?

10

How many numbers from 1 to 20, inclusive, are divisible by 3?

6

How many numbers from 1 to 20, inclusive, are divisible by 2 or 3?

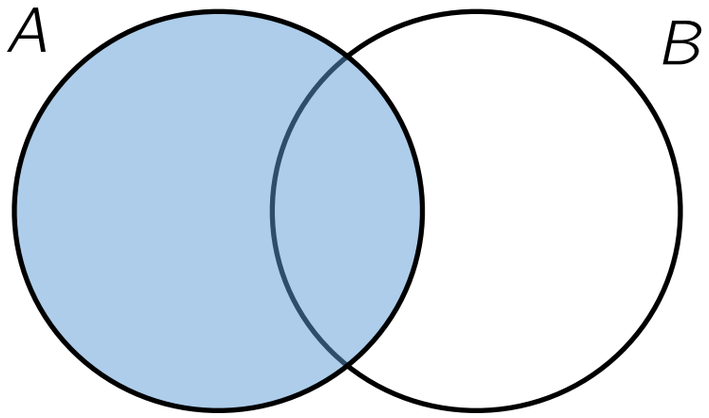
13

Convenient language: sets

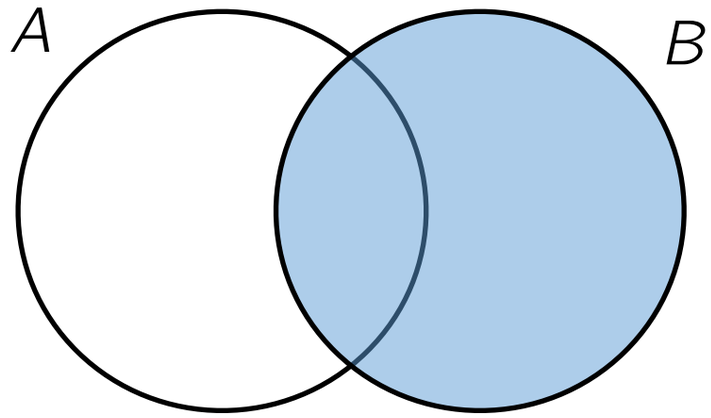
- Set is an arbitrary group of arbitrary elements
- Denote sets by A, B, C
- Order and repetitions of elements in the list is not important
- Sets can consist of anything

$$A = \{0, \sqrt{2}, \text{dog}\}$$

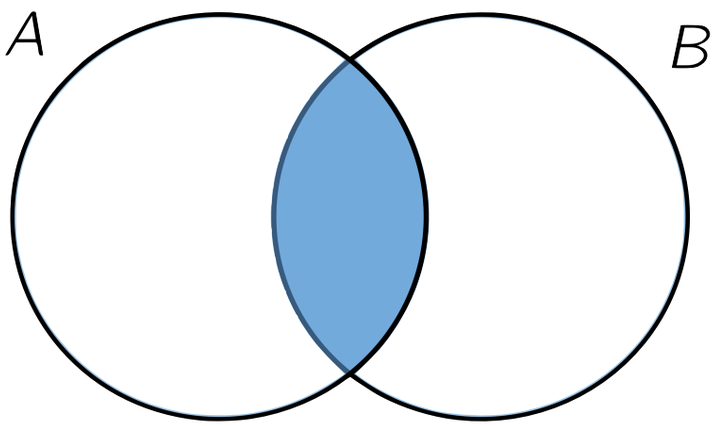
Venn Diagrams



Elements of A are within the left circle

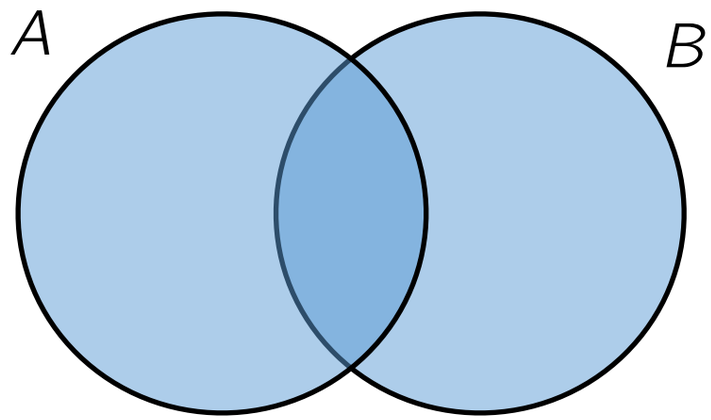


Elements of B are within the right circle



Elements belonging to both sets

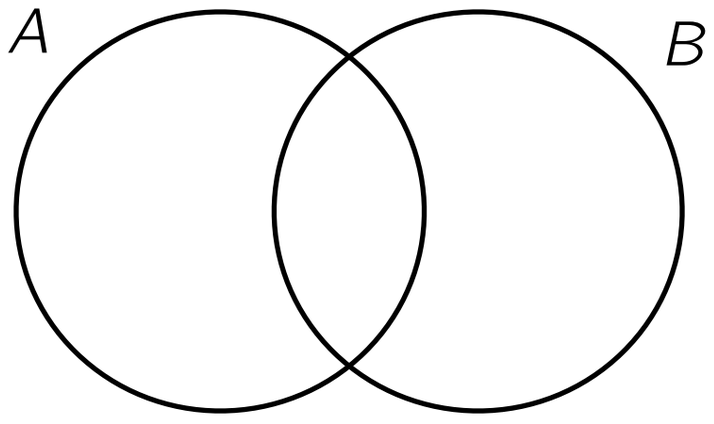
$A \cap B$



Elements belonging to at least one of the sets

$A \cup B$

Venn Diagrams

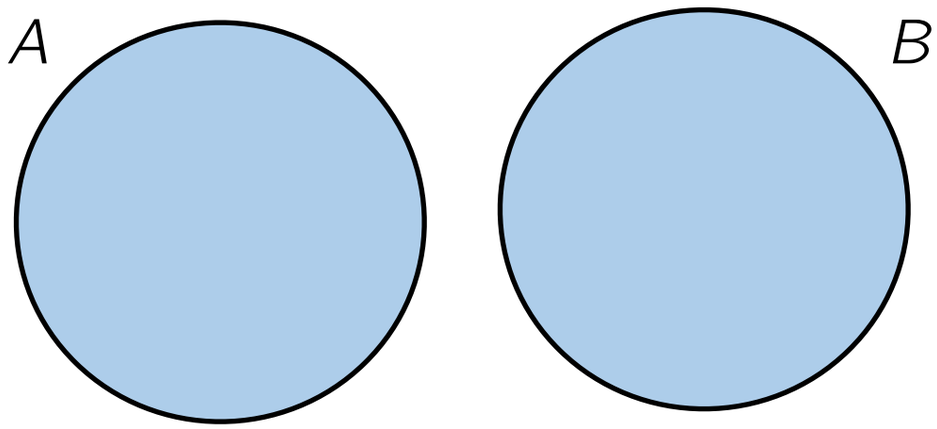


- If x is an element of A, we write $x \in A$
- The number of elements in A is $|A|$ (can be infinite)
- A set without elements is denoted by $\emptyset$ (called empty set)

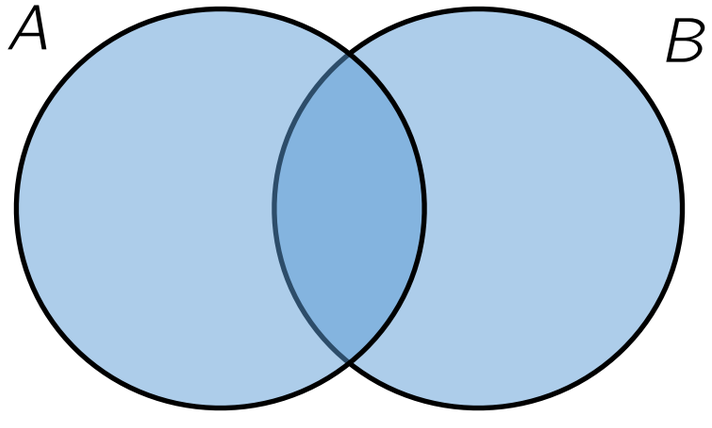
Rule of sum in the set language

Rule of sum

If there are k elements of a first type (in set A) and n elements of a second type (in set B), and these sets do not have common elements, then the set $A \cup B$ has $n+k$ elements of the one of two types



Generalized rule of sum



If we just consider $|A| + |B|$ as in the original rule, we would count elements that belong to both A and B twice!

Rule of sum

If there are k elements of a first type (in set A) and n elements of a second type (in set B), then

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

Exercise

Suppose we have 40 videos in our dataset. Each video falls in at least one of two categories, comedy videos or music videos. It is known that there are 27 comedy videos and 22 music videos in the dataset. How many videos fall into both categories?

7

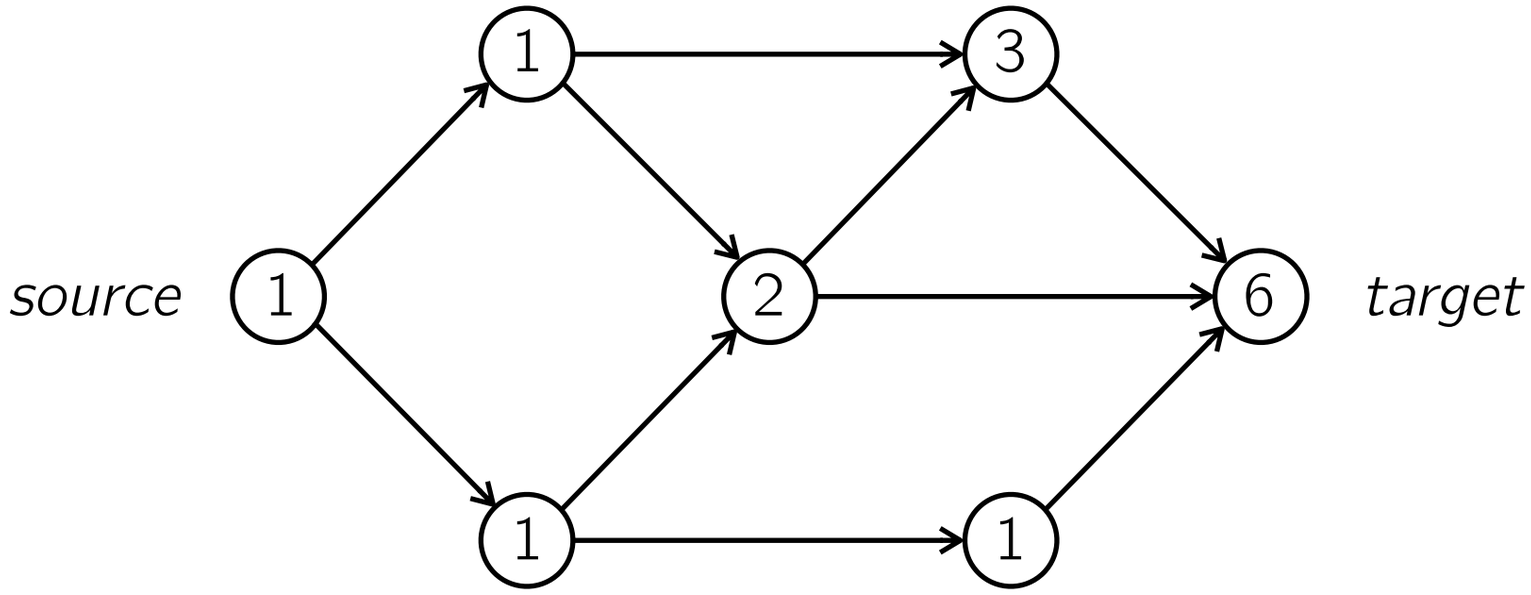
How many integer numbers from 1 to 1000 are divisible by 2 or by 3?

500 + 333 - 166 = 667

How many integer numbers from 1 to 1000 are not divisible neither by 2, nor by 3?

1000 - 667 = 333

Recursive counting: number of paths



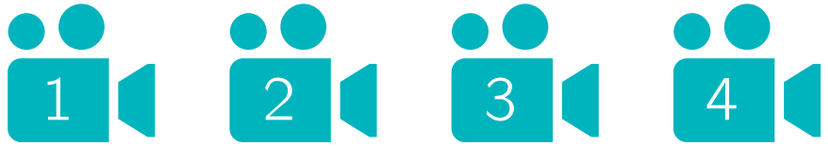
Consider nodes connected by edges on a direct graph. There is a starting node (called source) and a final node (called target). In how many ways can we traverse from the source node to the target node?

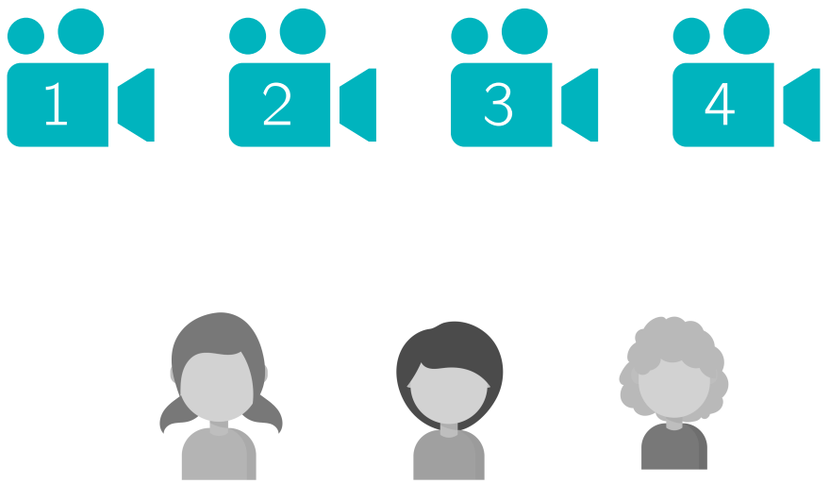
Idea:

- Count them recursively
- For each node count the number of paths from s to this node
- Use the rule of sum!

Rule of product

Rule of product
If there are k elements of the first type and there are n elements of the second type, then there are $k \times n$ pairs of elements, the first of the first type and the second of the second type





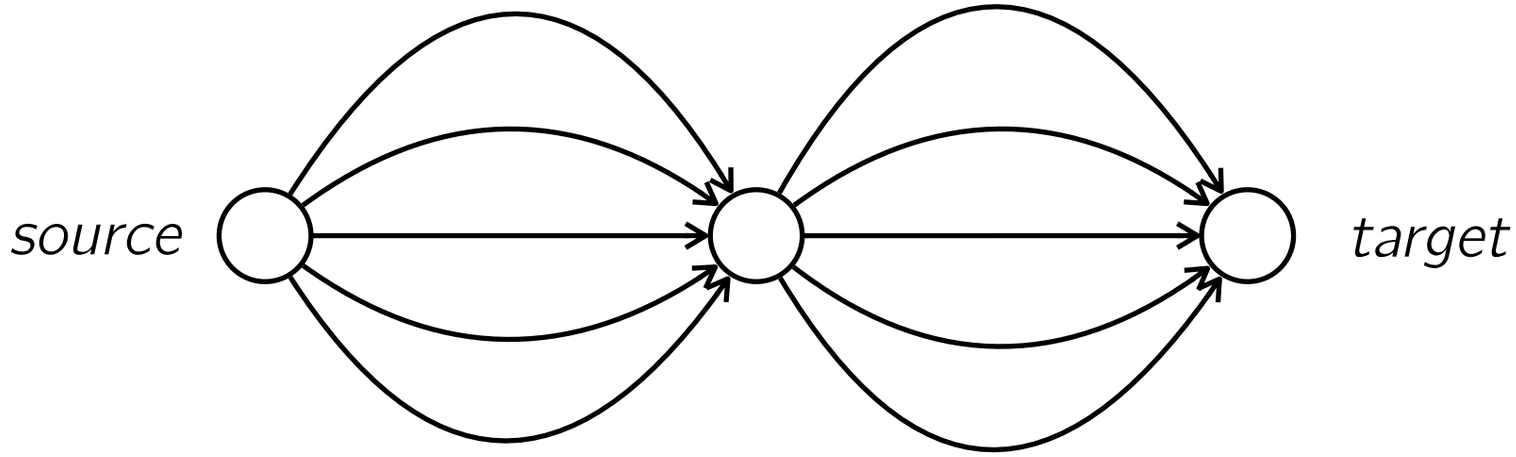
1	5	1	5
1	2	3	1
3	4	4	1

$4 \cdot 3 = 12$ evaluations

Rule of product in set language

Rule of product
If there is a finite set A and a finite set B, then there are $|A| \cdot |B|$ pairs of elements, the first from A and the second from B

Express the rule of product in terms of counting the number of paths?

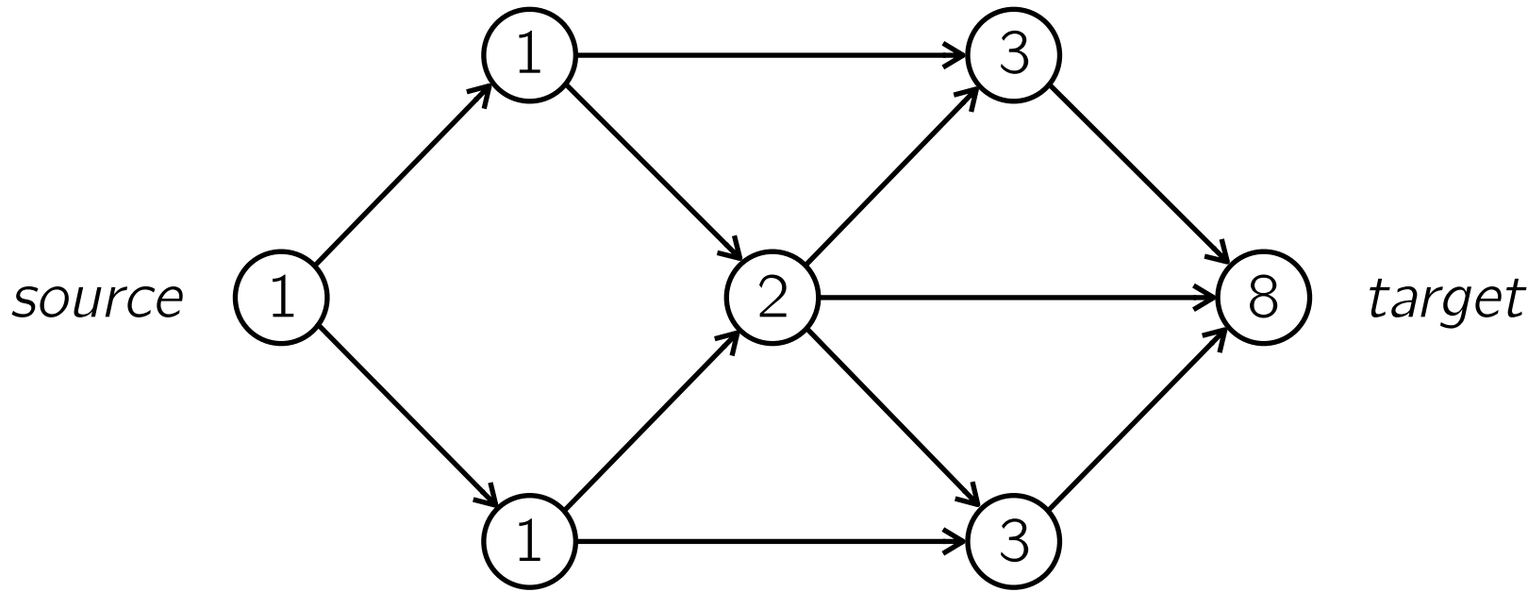


$|A| = 5$ $\qquad\qquad$ $|B| = 3$

$$|A| \cdot |B| = 15$$

Exercise

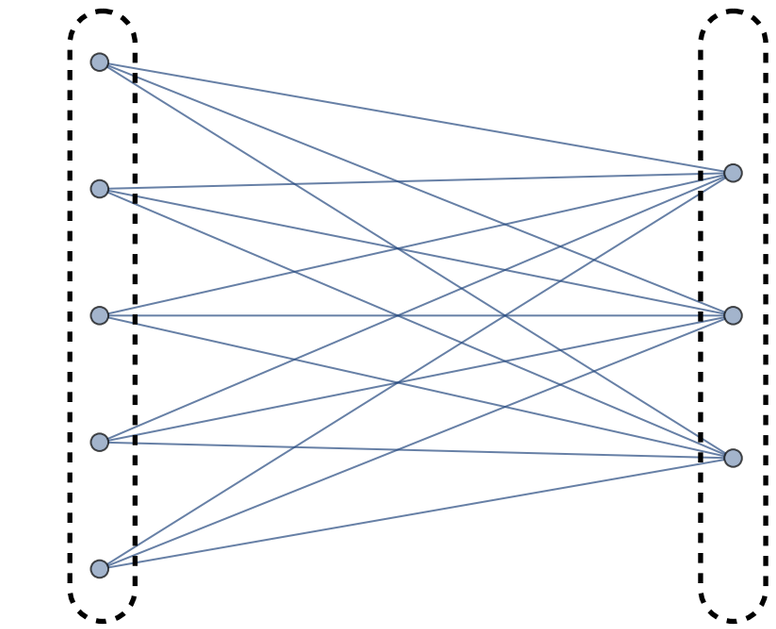
Consider the following set of points connected by arrows. How many different paths are there from source to target?



Suppose we have 7 disjoint datasets and each dataset contains 12 data entries. How many data entries do we have in total?

Exercise

What is the number of links in the undirected graph below?



$n = 5$ $m = 3$

Number of links:

$n \cdot m = 15$

What is the value of the variable count after running the following code?

```
count=0
for _ in range(4):
        for _ in range(8):
                count+=1
print(count)
```

Exercise

What is the value of the variable *count* after running the following code?

```
count=0
for _ in range(2):
        for _ in range(7):
                for _ in range(3):
                        count+=1
print(count)
```

What is the value of the variable *count* after running the following code?

```
count=0
for _ in range(4):
        for _ in range(4):
                count+=1
for _ in range(7):
        for _ in range(3):
                count+=1
print(count)
```

Summary

- Started by counting simple things
- Basic building blocks: rule of sum + rule of product
- How to take into account more sophisticated scenarios from these building blocks?