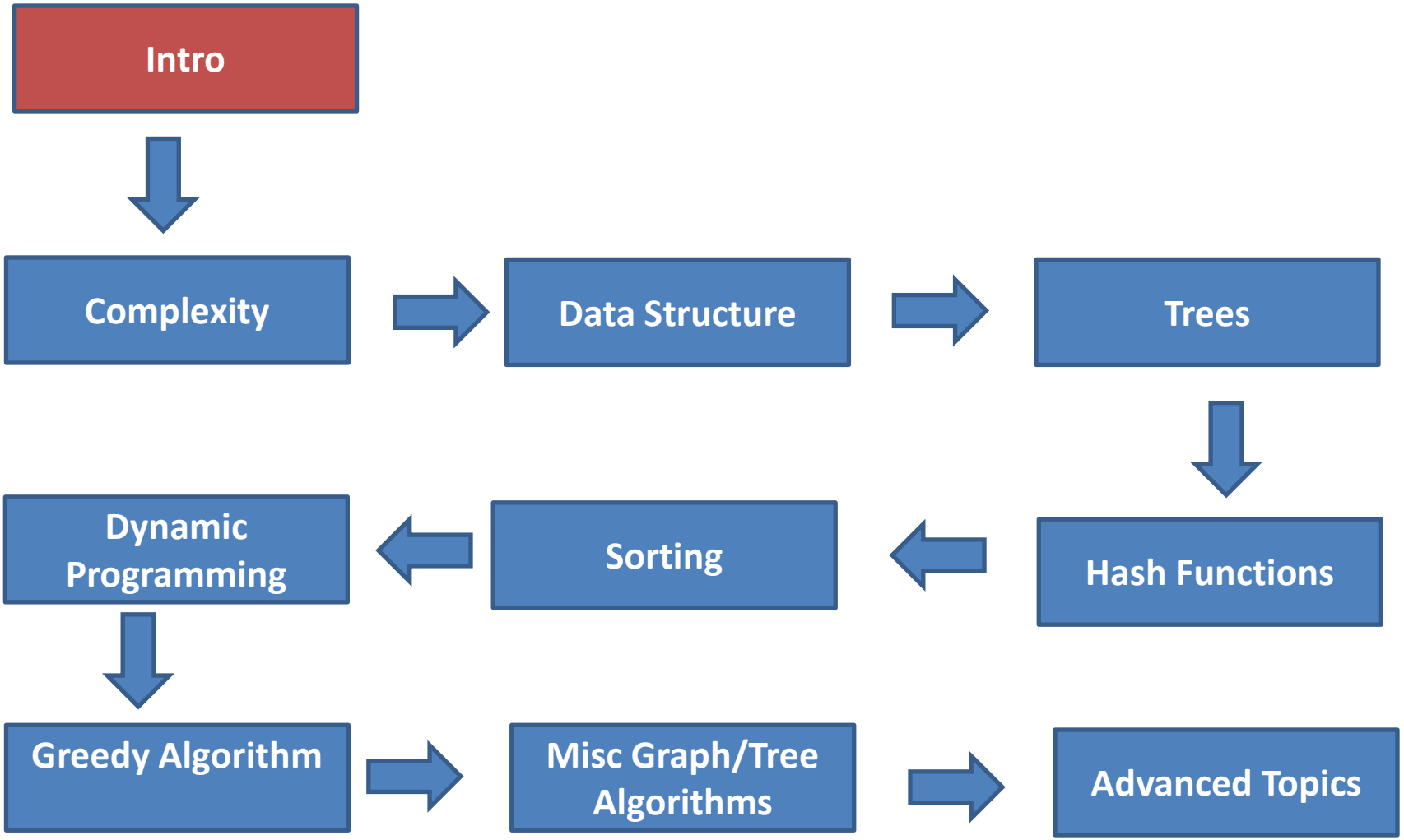
An Introduction to Algorithms

By

Hossein Rahmani

h_rahmani@iust.ac.ir

http://webpages.iust.ac.ir/h_rahmani/


Intro
Complexity
Data Structure
Trees
Hash Functions
Sorting
Dynamic Programming
Greedy Algorithm
Misc Graph/Tree Algorithms
Advanced Topics

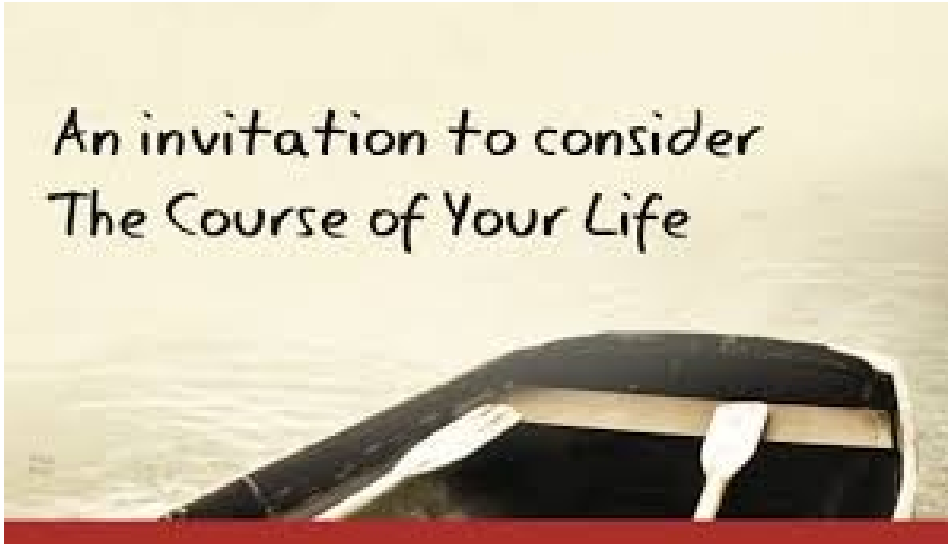
Why should I care about Algorithms?

Cartoon from *Intractability* by Garey and Johnson

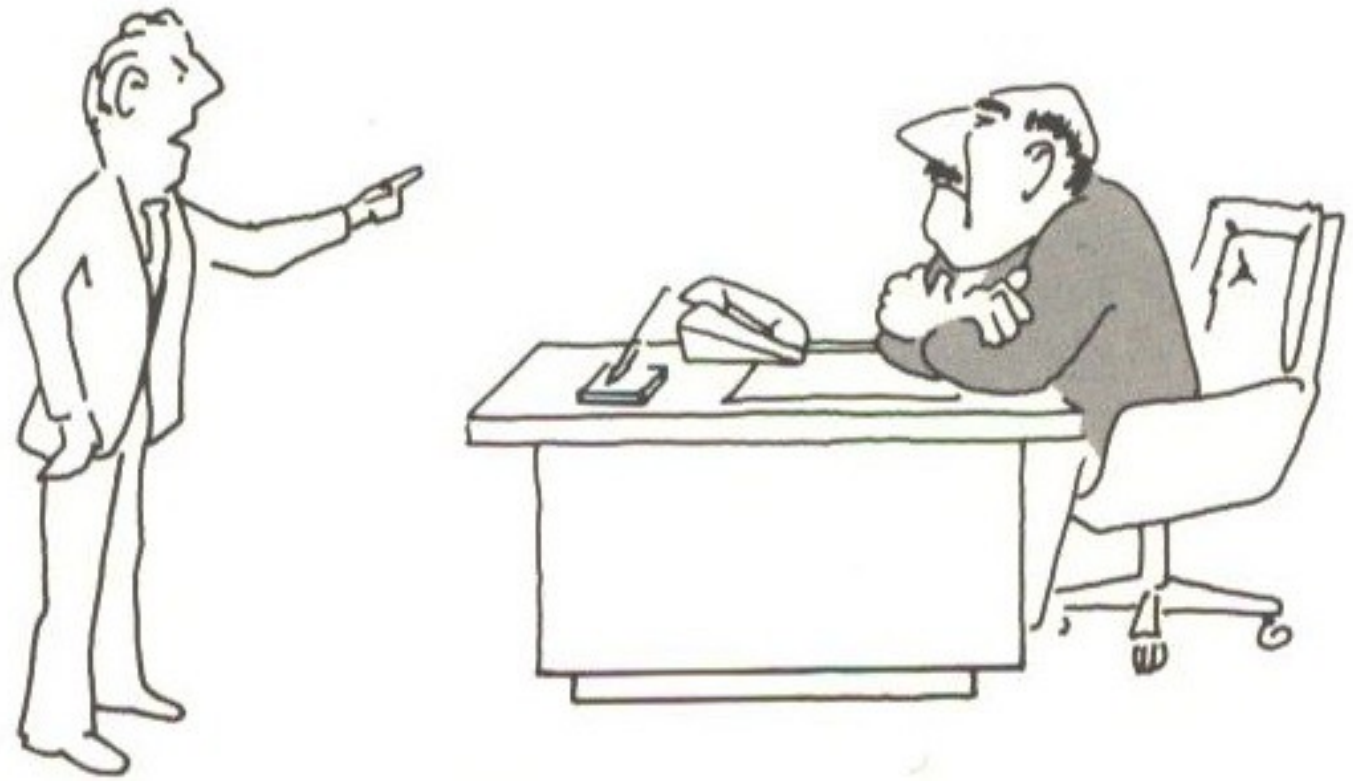
"I can't find an efficient algorithm, I guess I'm just too dumb."

More questions you should ask

- Who should know about Algorithms?
- Is there a future in this field?
- Would I ever need it if I want to be a software engineer or work with databases?



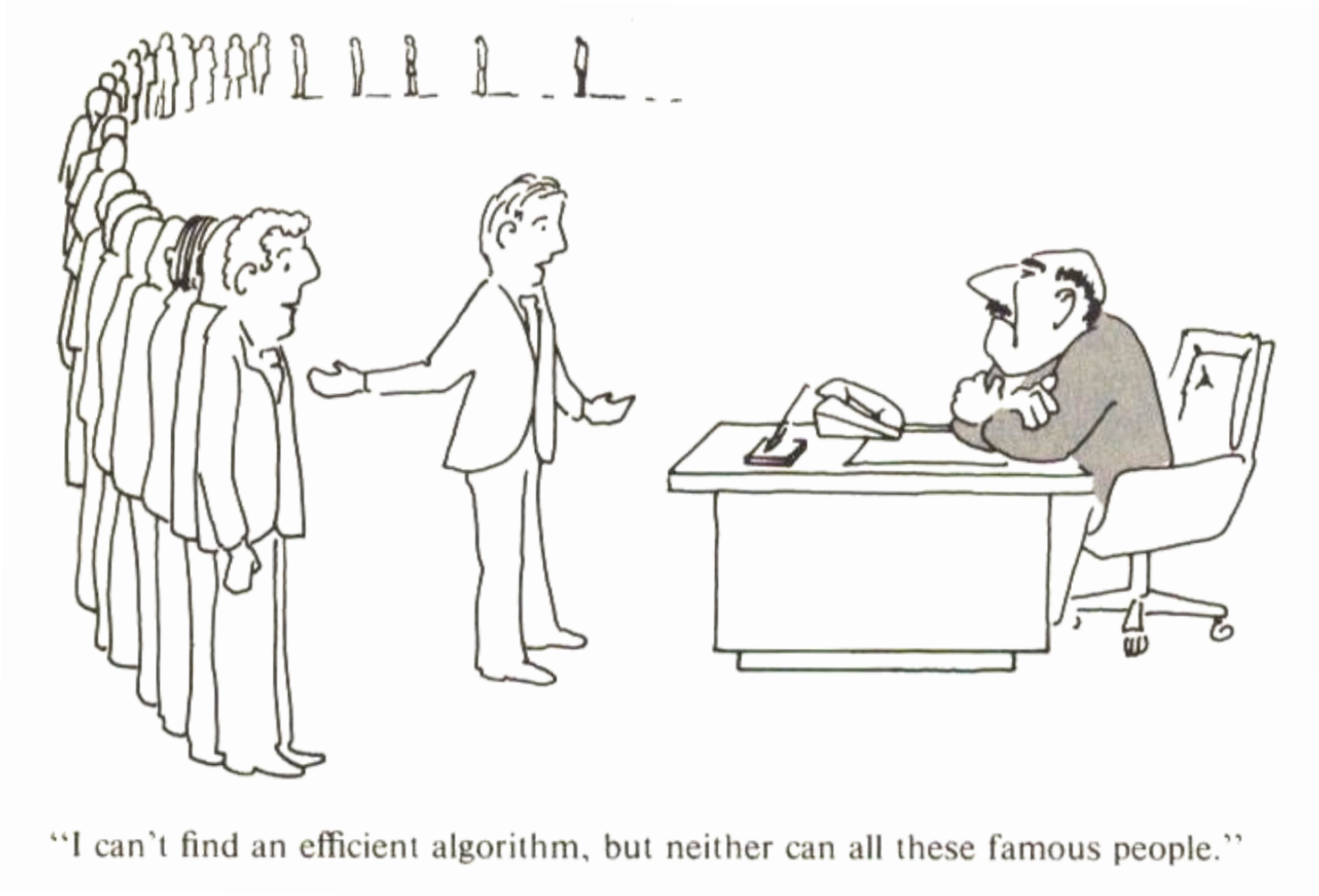
Why are theoretical results useful?



"I can't find an efficient algorithm, because no such algorithm is possible!"

Cartoon from *Intractability* by Garey and Johnson

Why are theoretical results useful?



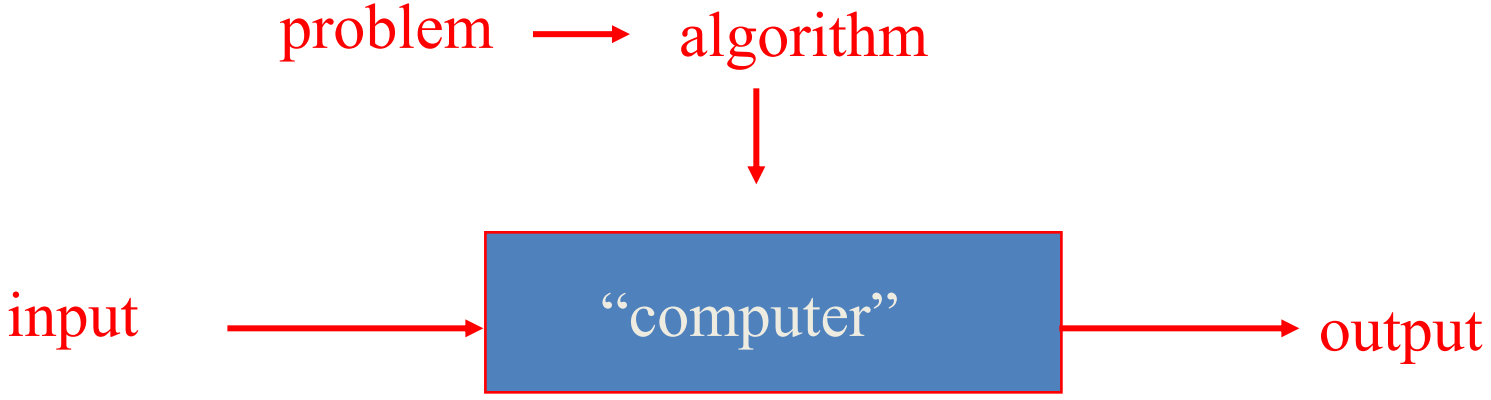
"I can't find an efficient algorithm, but neither can all these famous people."

Cartoon from *Intractability* by Garey and Johnson

Algorithm

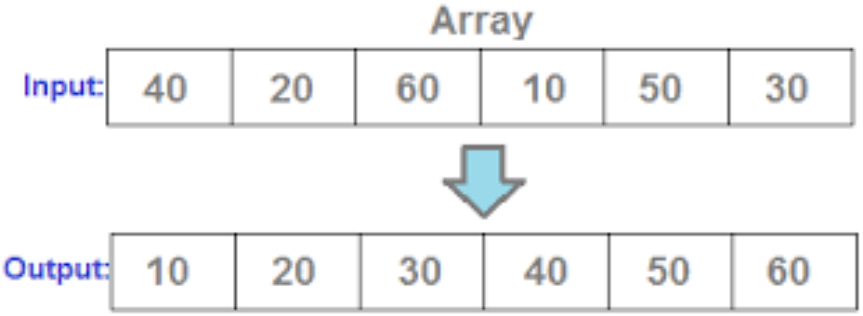
- A computational problem is a mathematical problem, specified by an input/output relation.

- An algorithm is a computational procedure for solving a computational problem.



Some Well-known Computational Problems

- Sorting
- Searching
- Shortest paths in a graph
- Traveling salesman problem
- Knapsack problem



Search

Text Data

Graph Data

Text Mining

What is Text Mining?

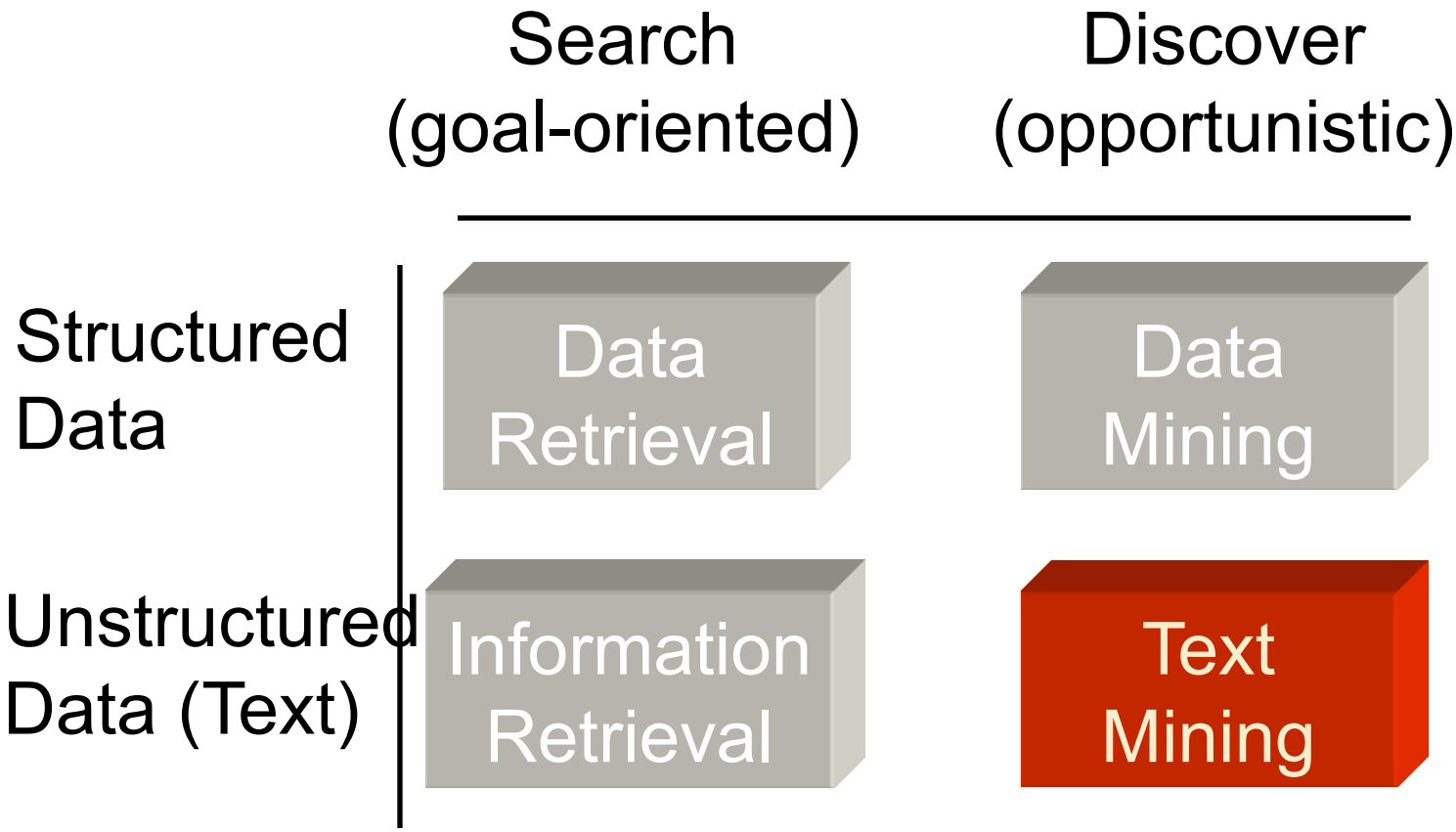
- There are many examples of text-based documents (all in 'electronic' format…)
  - e-mails, corporate Web pages, customer surveys, résumés, medical records, technical papers, incident reports, news stories and more…
- Not enough time or patience to read
  - Can we extract the most vital kernels of information…
- So, we wish to find a way to gain knowledge (in summarised form) from all that text, without reading or examining them fully first…!

What is Text Mining?

- Traditional data mining uses 'structured data' (n x p matrix)

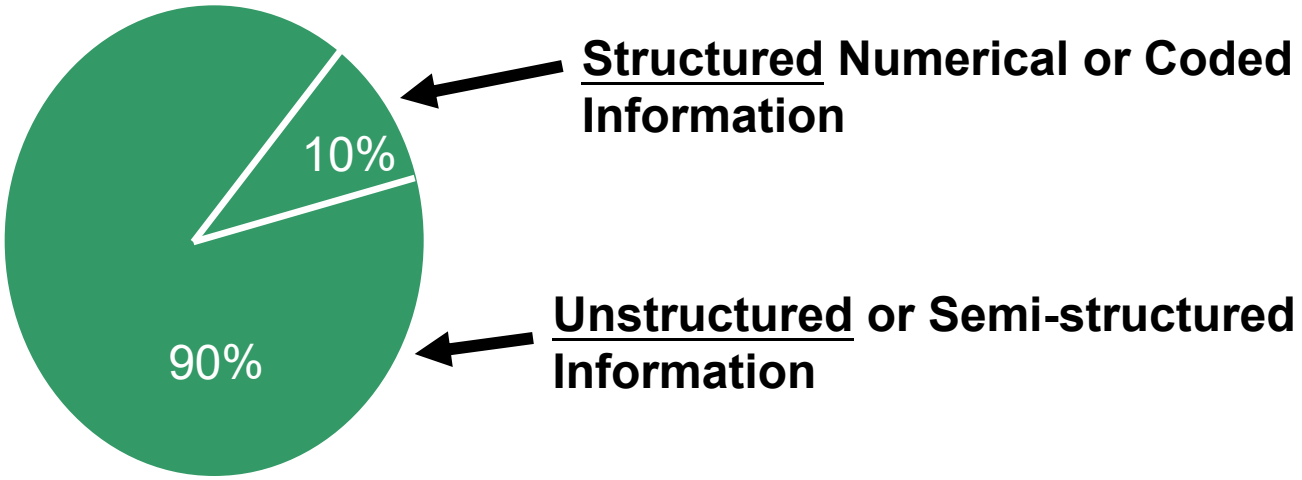
- The analysis of 'free-form text' is also referred to as 'unstructured data',
  - successful categorisation of such data can be a difficult and time-consuming task…

“Search” versus “Discover”



Motivation for Text Mining

- Approximately **90%** of the world's data is held in unstructured formats (source: Oracle Corporation)
- Information intensive business processes demand that we transcend from simple document retrieval to "knowledge" discovery.



Text Mining: Examples

- Text mining is an exercise to gain knowledge from stores of language text.
- Text:
  - Web pages
  - Medical records
  - Customer surveys
  - Email filtering (spam)
  - Incident reports
  - Drug interaction reports
  - News stories (e.g. predict stock movement)

Text Mining

- Typically falls into one of two categories
  - Analysis of text: I have a bunch of text I am interested in, tell me something about it
    - E.g. sentiment analysis, “buzz” searches
  - Retrieval: There is a large corpus of text documents, and I want the one closest to a specified query
    - E.g. web search, library catalogs, legal and medical precedent studies

Text Mining: Analysis

- Which words are most present
- Which words are most surprising
- Which words help *define* the document
- What are the interesting text phrases?

Text Mining: Retrieval

- Find k objects in the corpus of documents which are most similar to my query.
- Can be viewed as "interactive" data mining - query not specified a priori.
- Main problems of text retrieval:
  - What does "similar" mean?
  - How do I know if I have the right documents?
  - How can I incorporate user feedback?

Text Retrieval: Challenges

- Calculating similarity is not obvious - what is the distance between two sentences or queries?
- Evaluating retrieval is hard: what is the "right" answer ? (no ground truth)
- User can query things you have not seen before e.g. misspelled, foreign, new terms.
- Goal (score function) is different than in classification/regression: not looking to model all of the data, just get best results for a given user.
- Words can hide semantic content
  - Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

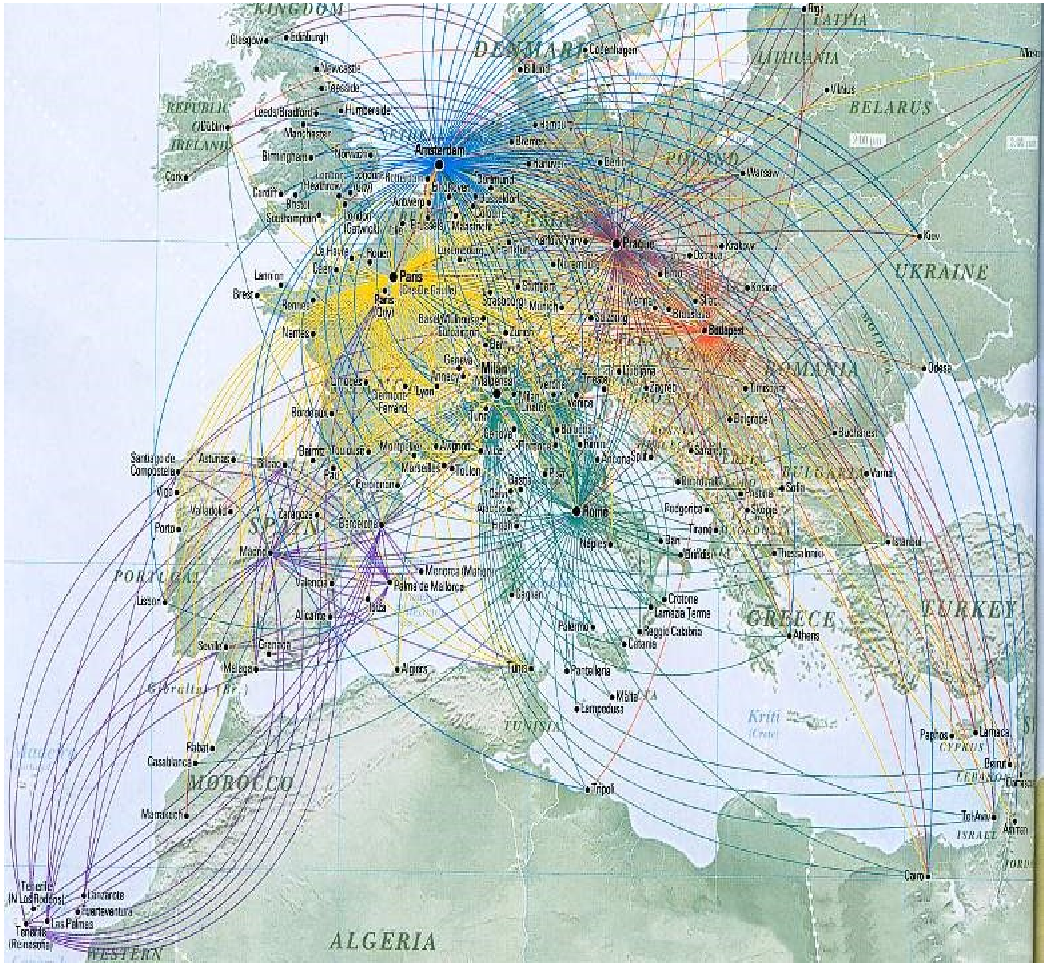
Graph Data

Many examples of networks

- Technological Networks
- Social Networks
- Networks of Information
- Biological Networks
- And …

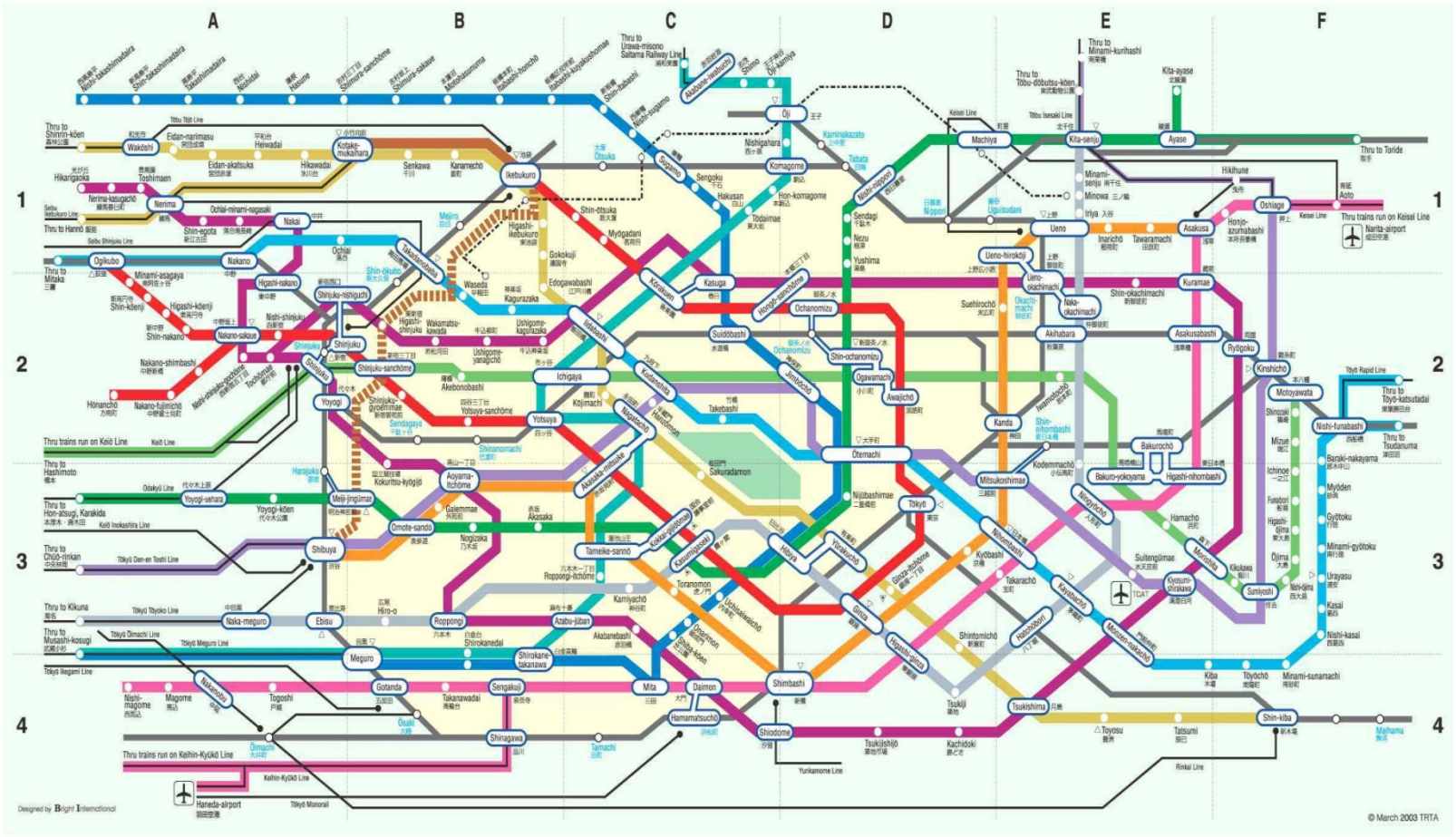
TECHNOLOGICAL NETWORKS

The Airline Networks



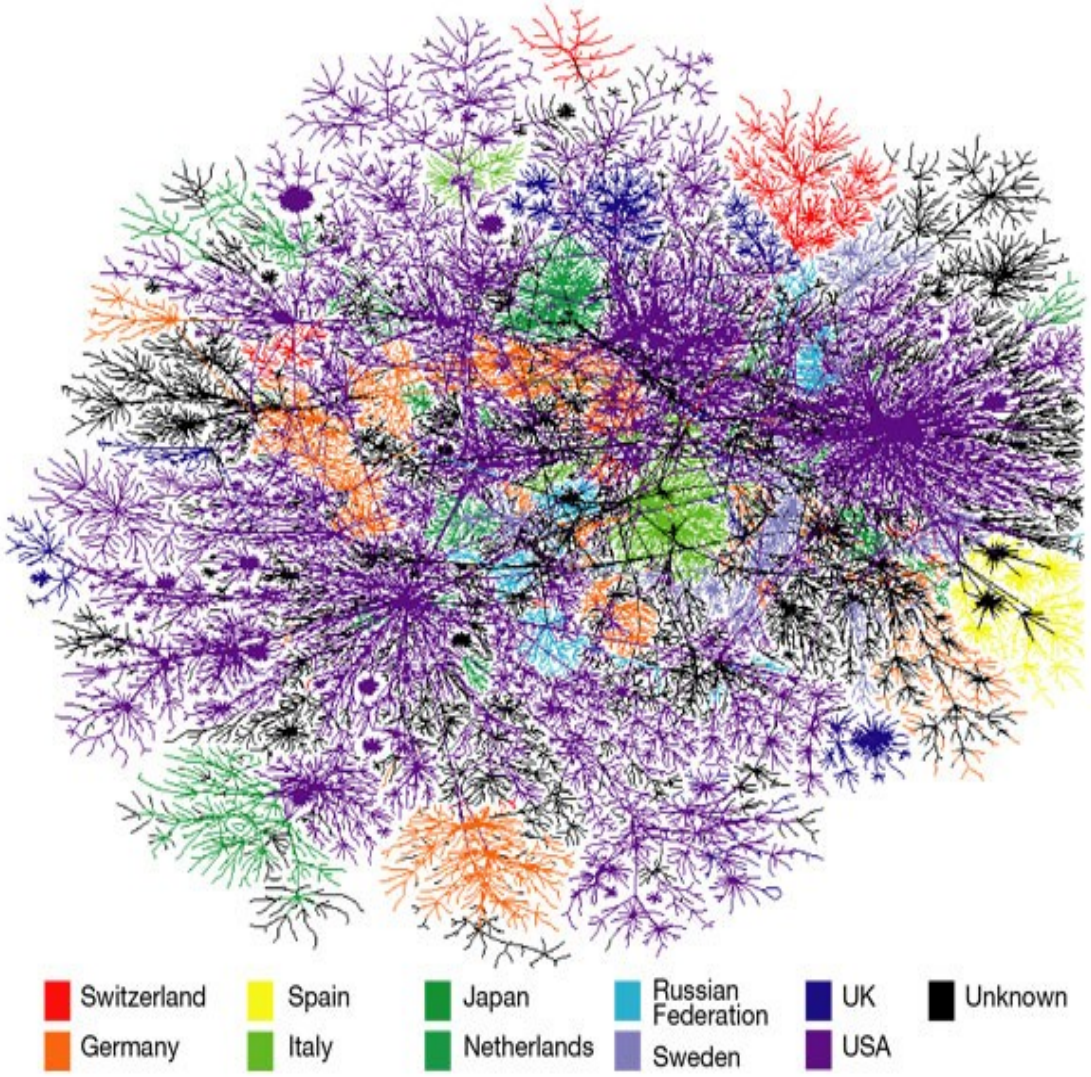
Quiz Grade =1/number of people with similar answers

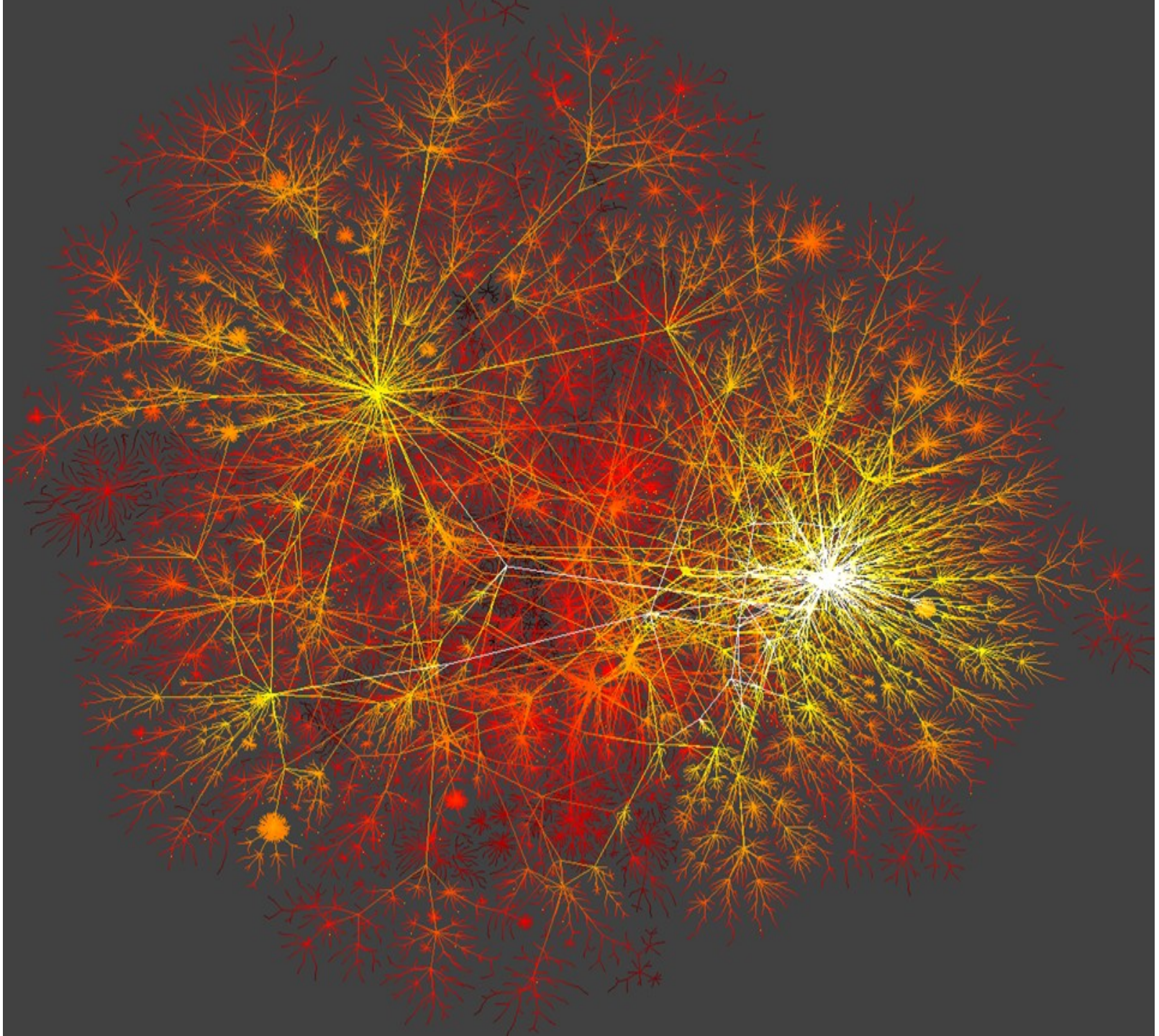
Railway Networks



Source: TRTA, March 2003 - Tokyo rail map

The Internet Map



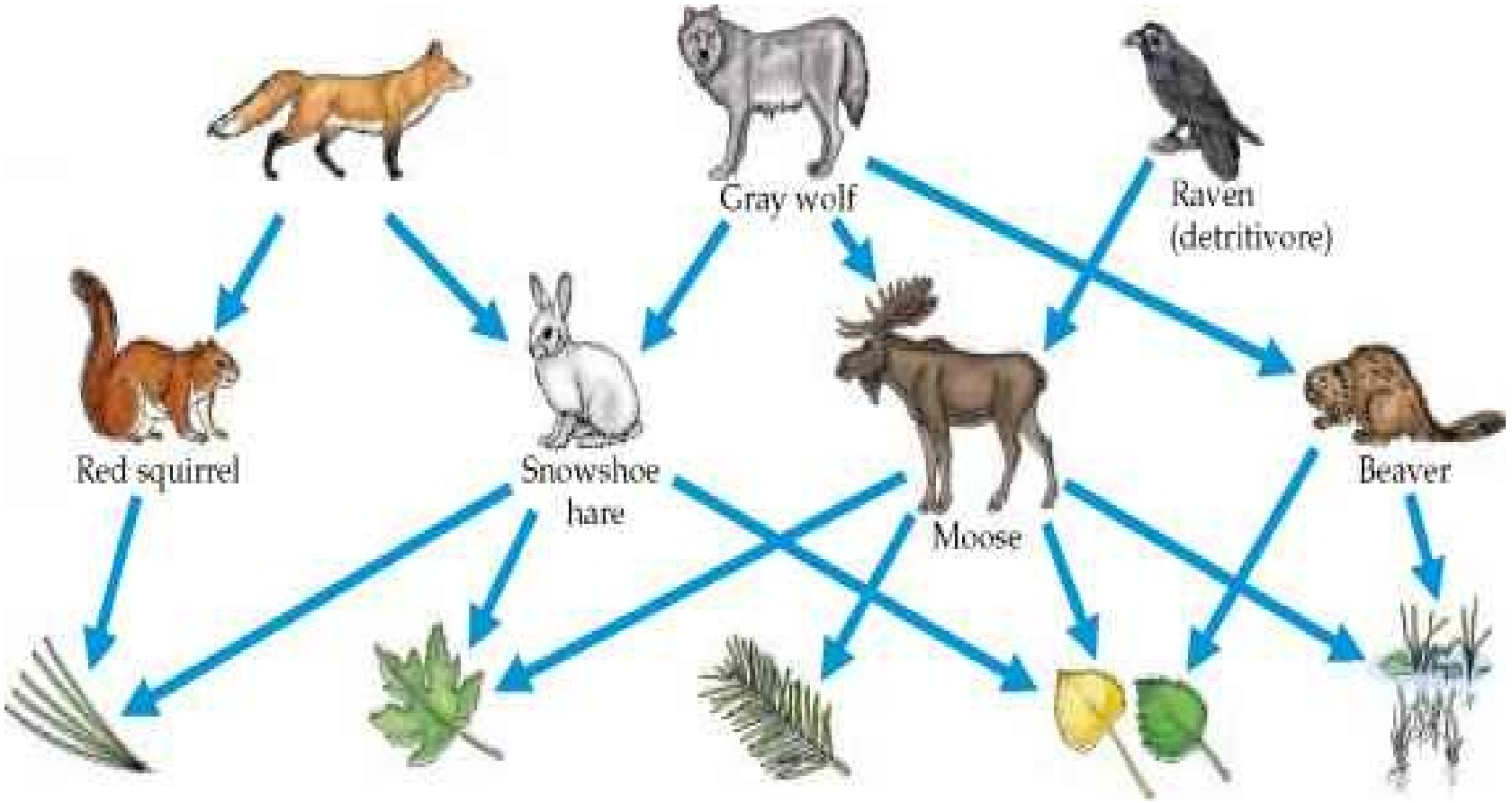


Other Technological Networks?

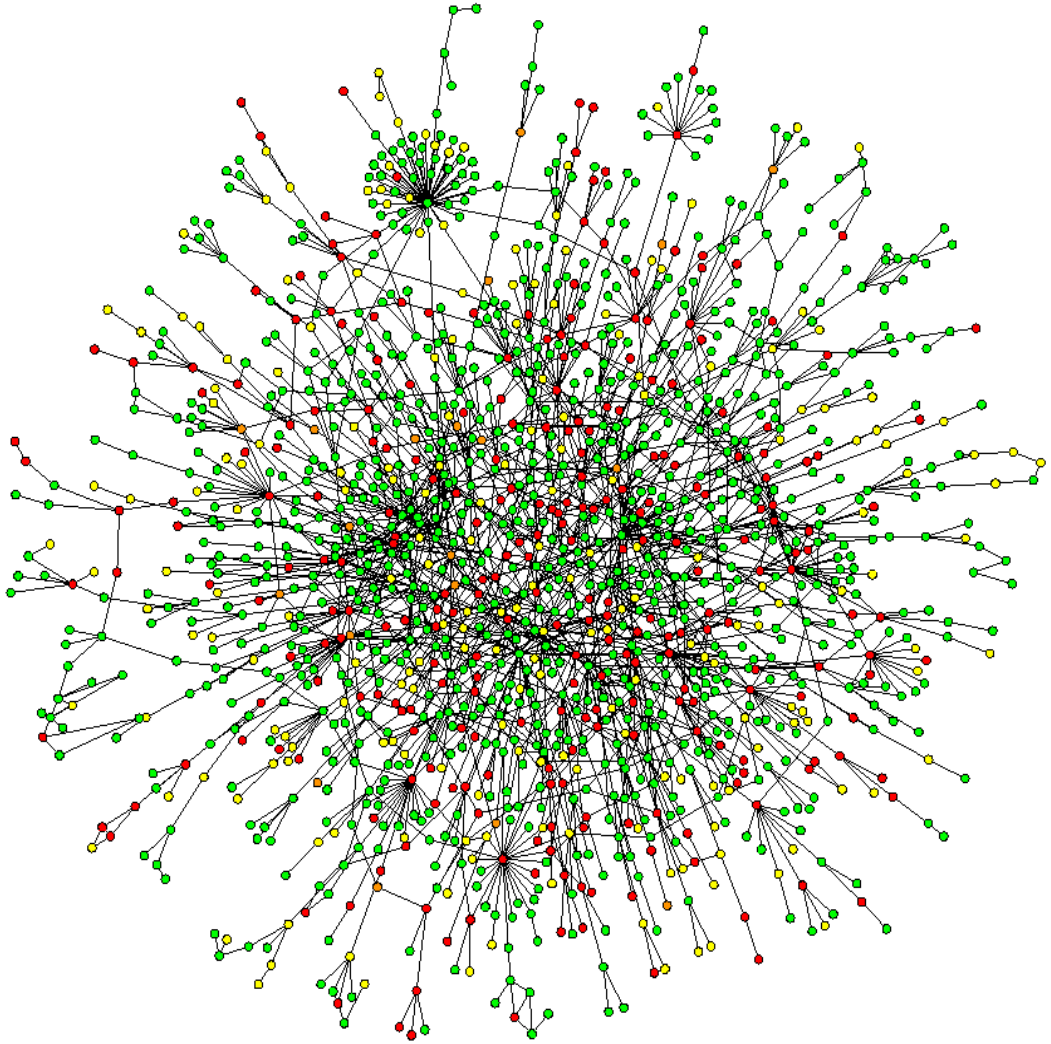
- Internet
- Telecommunication Networks, e.g., telephone network
- Power Grid
  - The network of high-voltage transmission lines
  - that provide long-distance transport of electric power
- Transportation networks
  - Airlines, Railway, …
- Delivery and distribution networks
  - Gas, oil, water, Post, …

BIOLOGICAL NETWORKS

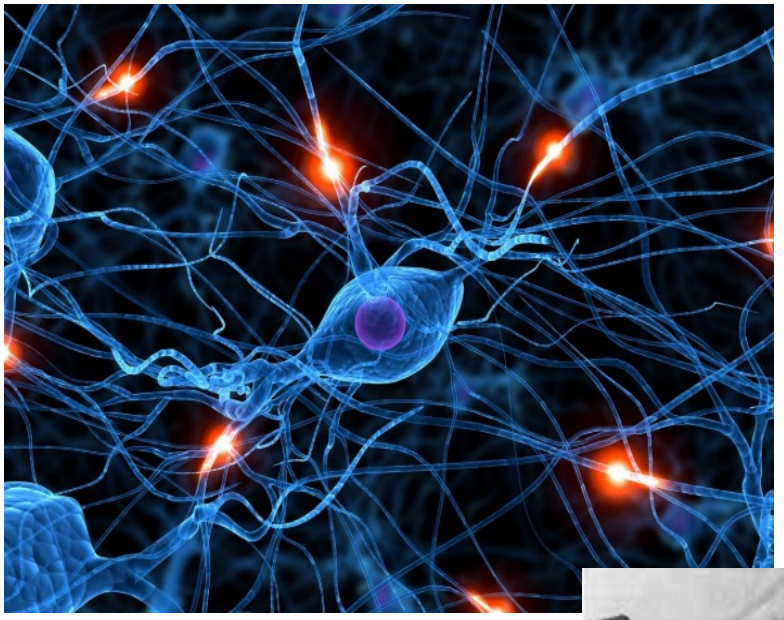
Food Webs (Ecology)

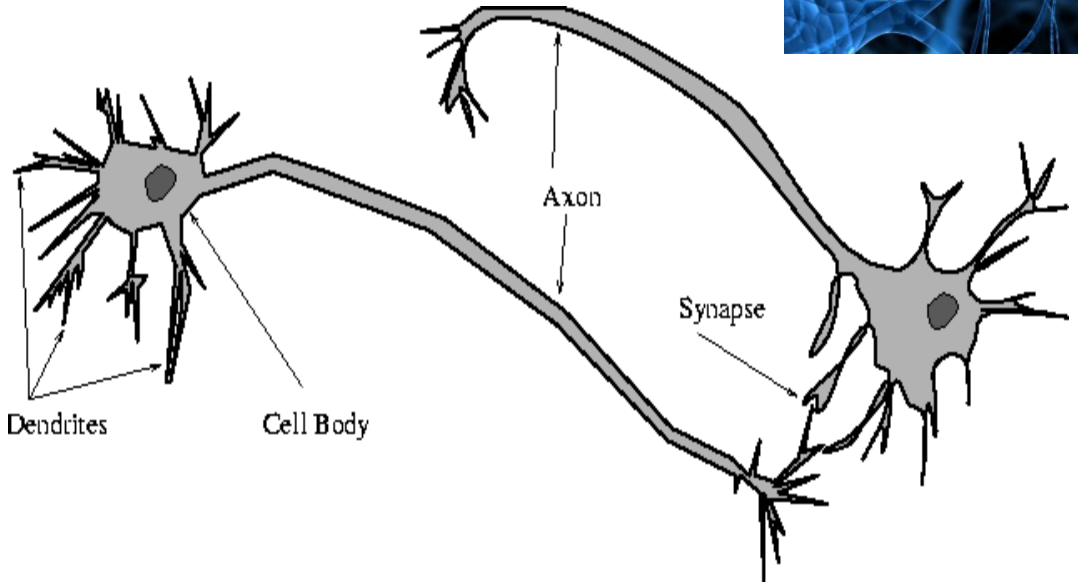


Protein-Protein Interaction

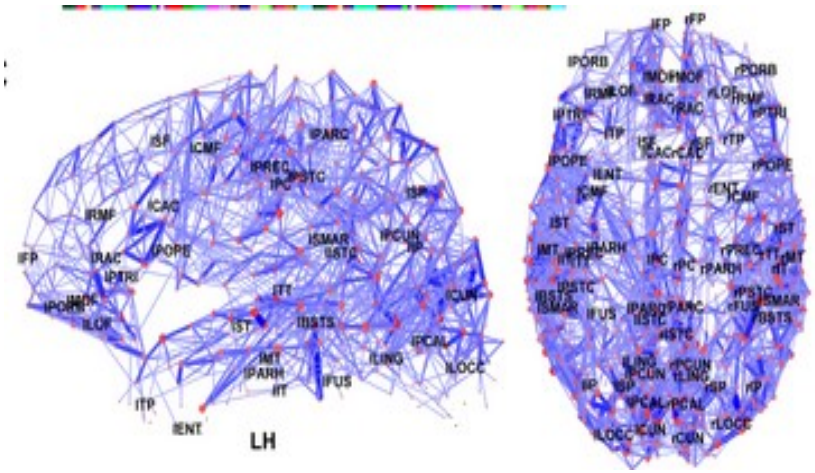


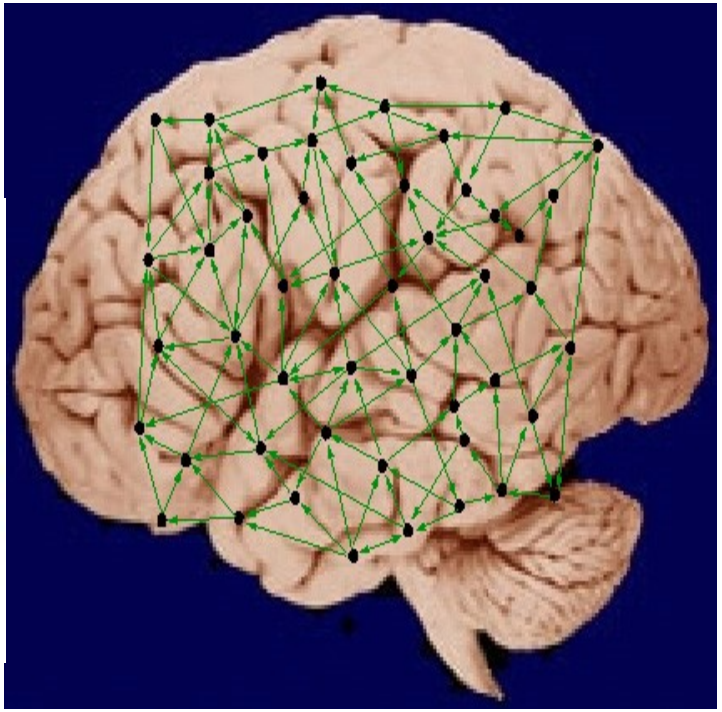
Neural Networks





Brain Networks



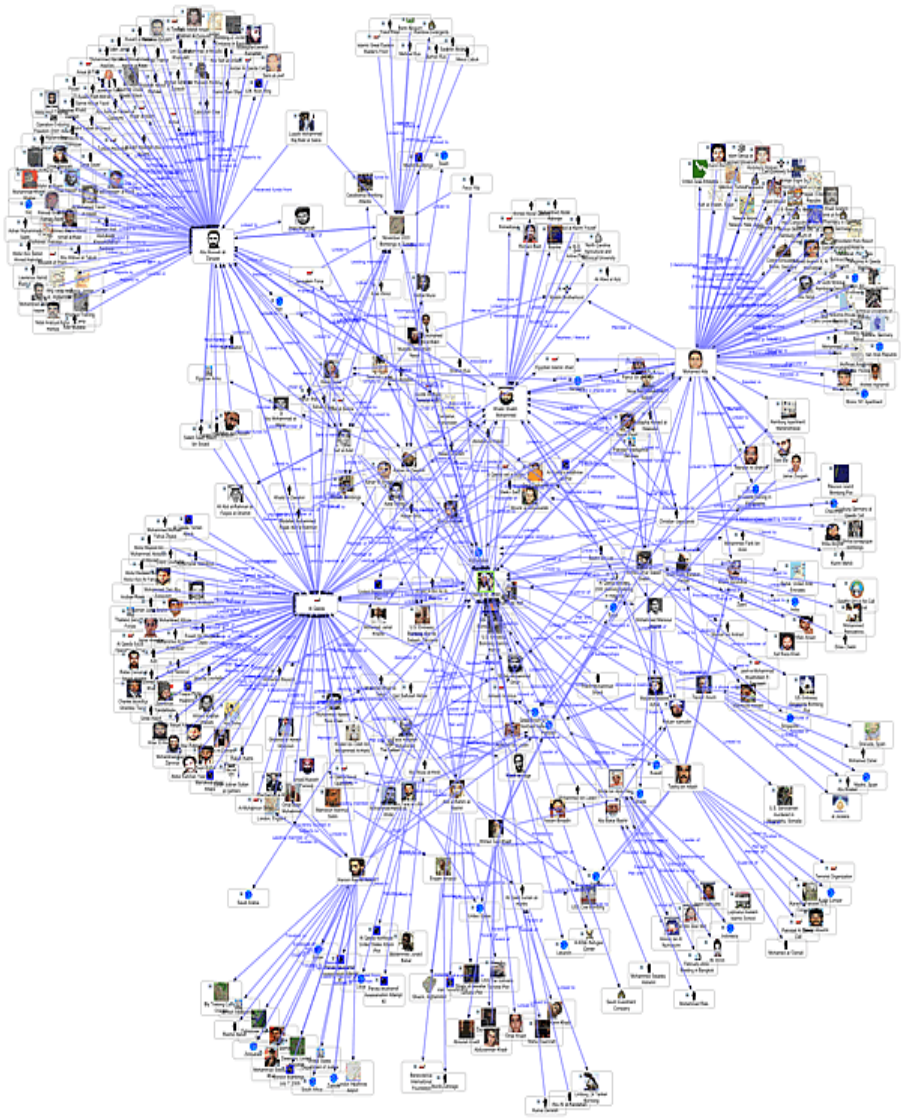


Other Biological Networks

- Metabolic Networks
- Gene Regulatory Network
- Metabolic Pathways

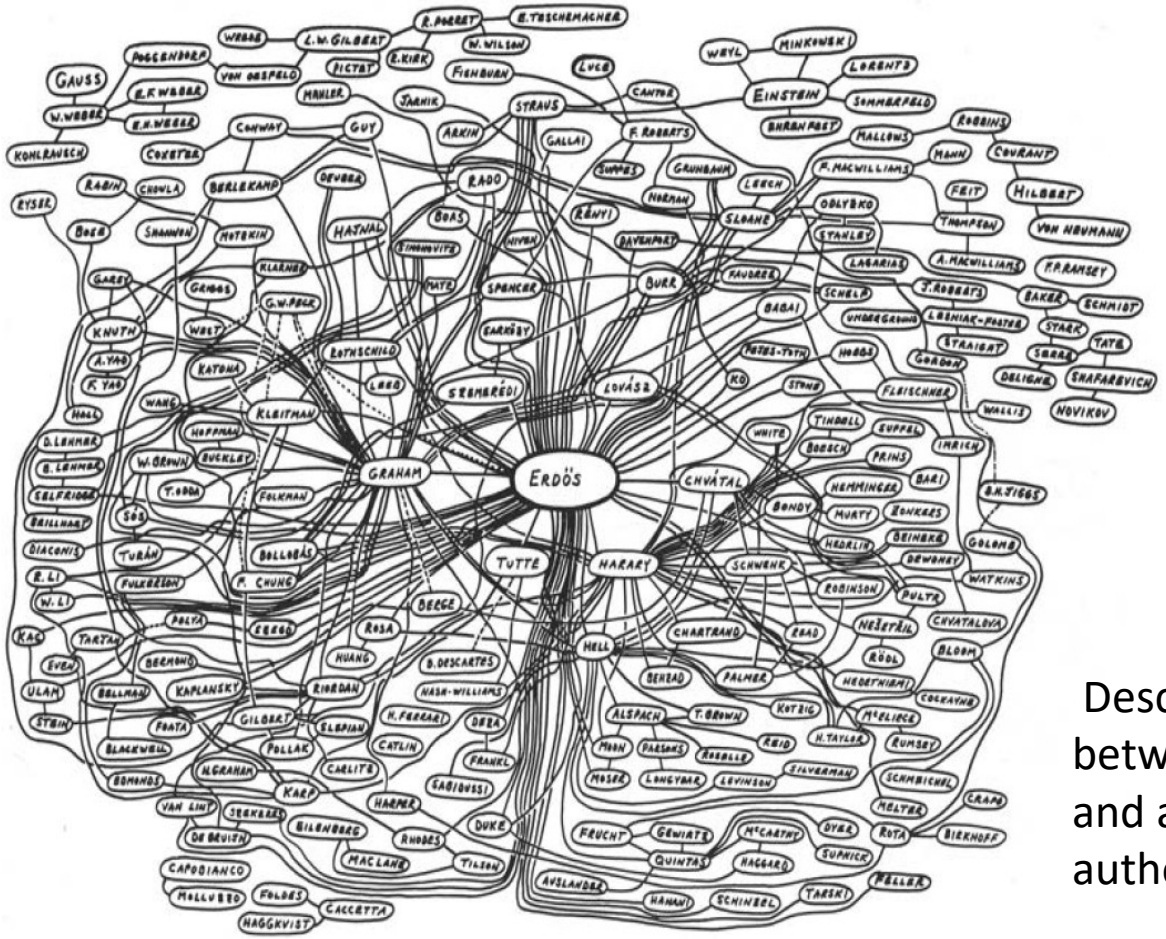
SOCIAL NETWORKS

Friendship Networks



Online Social Networks

Co-authorship Network

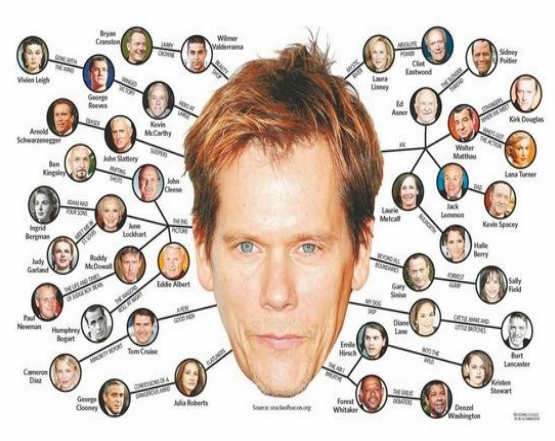


Erdős number

Describes the "collaborative distance" between mathematician Paul Erdős and another person, as measured by authorship of mathematical papers.

Co-stardom Networks

- the collaboration graph of film actors
- Six Degrees of *Kevin Bacon*

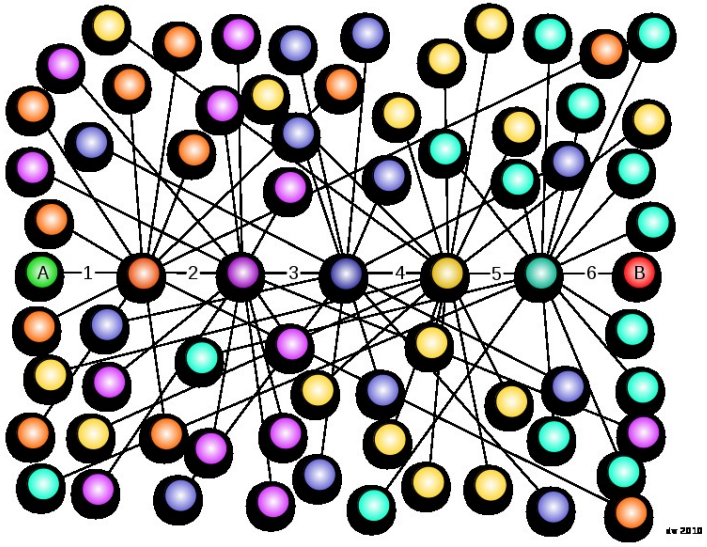


- Who is the co-star hub of Iranian films?!

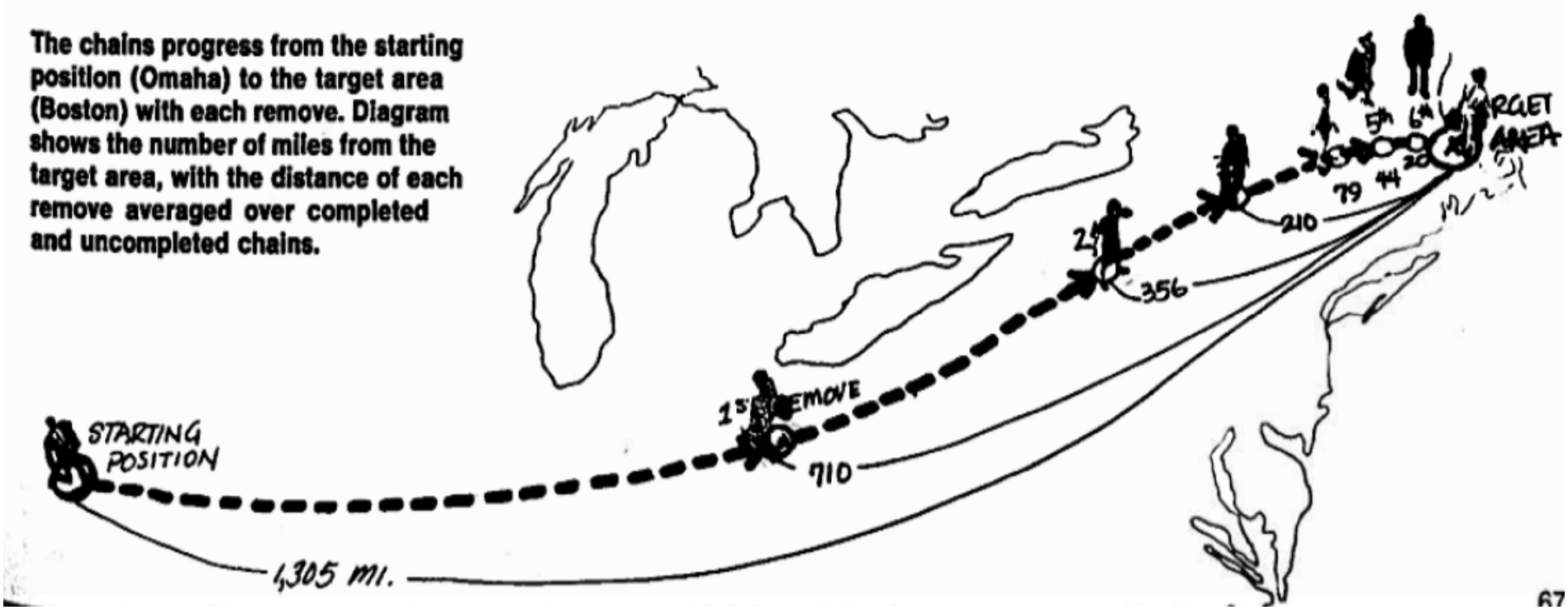


Six Degrees of *Kevin Bacon*

- **Six degrees of separation**
  - *All living things and everything else in the world is six or fewer steps away from each other so that a chain of "a friend of a friend" statements can be made to connect any two people in a maximum of six steps.*



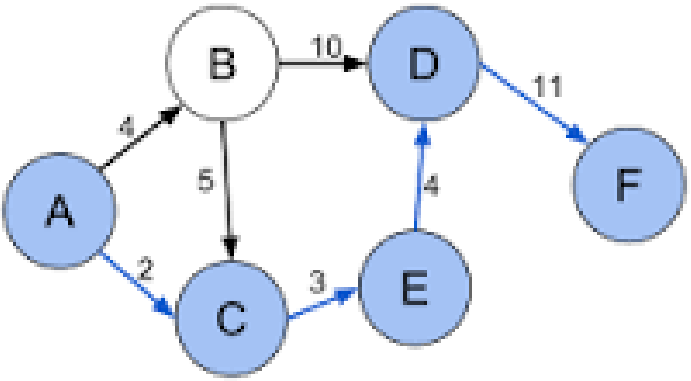
Milgram Experiment

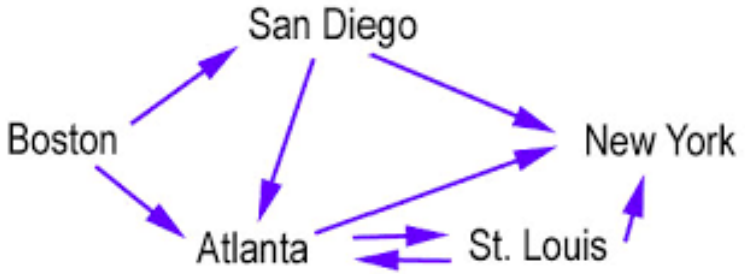


Milgram's Experiment

- It demonstrated two striking facts about large social networks:
  - 1- short paths are there
  - 2- people are effective at collectively finding these short paths
    - Acting without any sort of global "map" of the network,
- It is possible: a social network where the first true but the second isn't
  - a letter might wander from one acquaintance to another, lost in a maze of social connections

Shortest paths in a graph

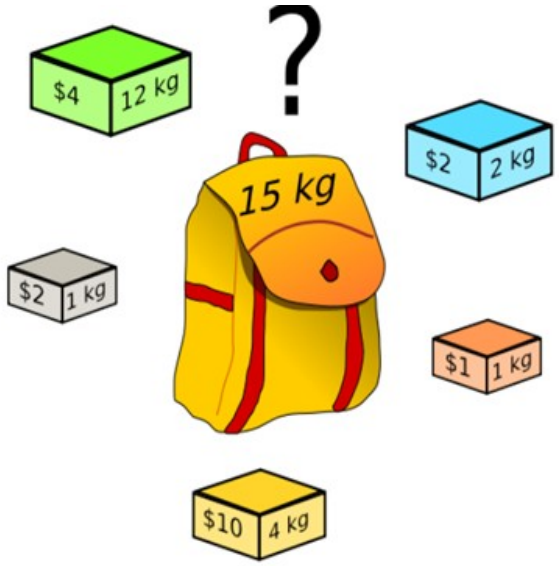




Travel Salesman Problem

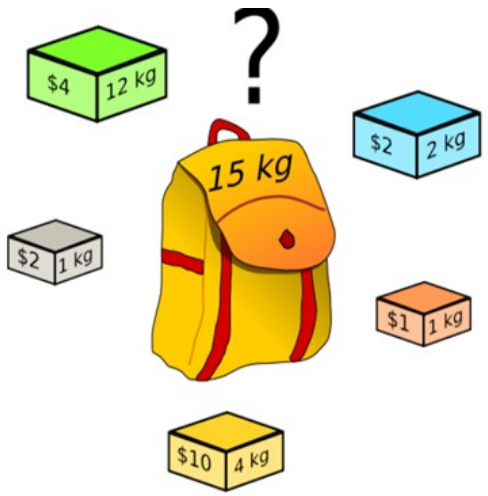
Knapsack problem

- You have a knapsack that has the capacity (weight) W
- You have several Items
- Each item has weight wi and weight bj
- You want to put items in the knapsack so that
  - Knapsack capacity is not exceeded
  - Total benefits maximizes



Knapsack problem Types

- 0-1 knapsack problem
  - Items can not be divided
- Fractional knapsack problem
  - For instance, items are liquid or powder



The great thinkers of our field

Euclid

Euclid by Justus van Gent, 15th century

Born	Mid-4th century BCE
Died	Mid-3rd century BCE
Residence	Alexandria, Hellenistic Egypt
Fields	Mathematics
Known for	Euclidean geometry Euclid's *Elements* Euclidean algorithm

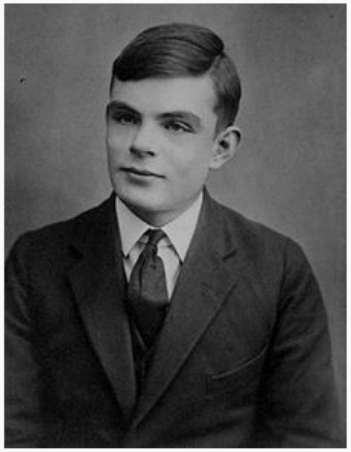
Muḥammad ibn Mūsā al-Khwārizmī

A stamp issued September 6, 1983 in the Soviet Union, commemorating al-Khwārizmī's (approximate) 1200th birthday.

Born	c. 780 Khwarezm[1]
Died	c. 850
	Academic work
Era	Medieval era (Islamic Golden Age)
Notable ideas	Treatises on algebra and Indian numerals
Influenced	Abu Kamil[2]

Alan Turing

OBE FRS



Turing aged 16

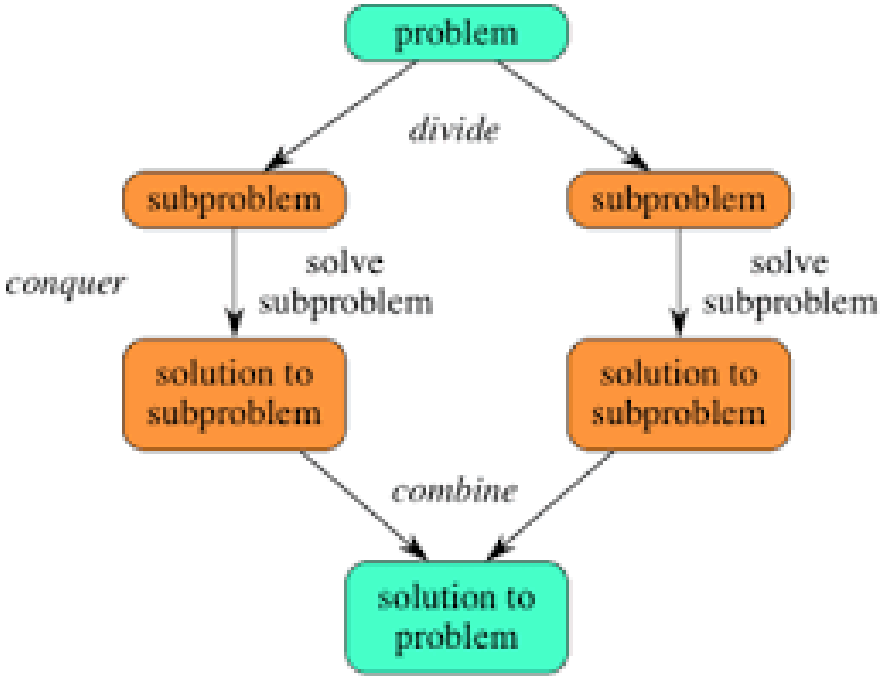
Born	Alan Mathison Turing 23 June 1912 Maida Vale, London, England
Died	7 June 1954 (aged 41) Wilmslow, Cheshire, England
Residence	Wilmslow, Cheshire, England
Citizenship	United Kingdom
Fields	Mathematics, cryptanalysis, logic, computer science, mathematical and theoretical biology

Basic Issues Related to Algorithms

- How to design algorithms
- How to express algorithms
- Proving correctness
- Efficiency
  - Theoretical analysis
  - Empirical analysis
- Optimality

Algorithm design strategies

- Brute force (Proof by exhaustion, proof by cases, proof of each of the cases)
- Divide and conquer
- Greedy approach
- Dynamic programming
- Backtracking
- Branch and bound
- Space and time tradeoffs



Analysis of Algorithms

- How good is the algorithm?
  - Correctness
  - Time efficiency: amount of work done
  - Space efficiency: amount of space used
  - Simplicity, clarity
- Does there exist a better algorithm?
  - Lower bounds
  - Optimality

Correctness

Proving correctness is dreadful for large algorithms. A strategy that can be used is: divide the algorithm into smaller pieces, and then clarify what the preconditions and postconditions are and prove correct assuming everything else is correct.

Amount of Work Done

Rather than counting the total number of instructions executed, we'll focus on a set of key instructions and count how many times they are executed. Use asymptotic notation and pay attention only to the largest growing factor in the formula of the running time.

Two major types of analysis: worst-case analysis and average-case analysis

More

- Amount of space used: The amount of space used can be measured similarly. Consideration of this efficiency is often important.

- Simplicity, clarity: Sometimes, complicated and long algorithms can be simplified and shortened by the use of recursive calls.

- Optimality: For some algorithms, you can argue that they are the *best* in terms of either amount of time used or amount of space used. There are also problems for which you cannot hope to have efficient algorithms.

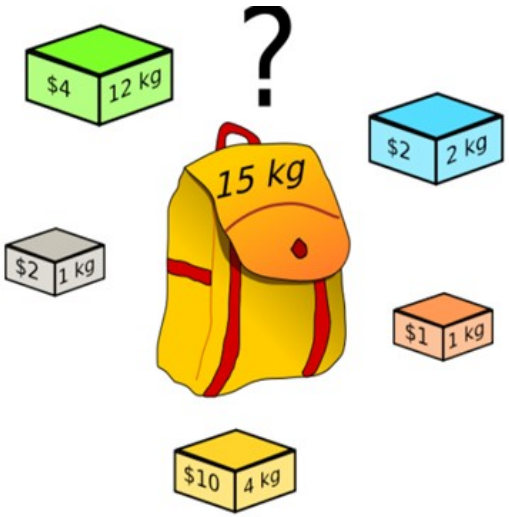
Quiz 1

- Given a polynomial
  - $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n$

  compute the value of the polynomial for a given value of x.
- How many additions and multiplications are needed?

Quiz 2

- Propose a algorithm for solving a Fractional knapsack problem.



Quiz 1 Solution

- Simple solution:
  - Number of additions = n
  - Number of multiplications = 1 + 2 + … + n = n(n+1)/2