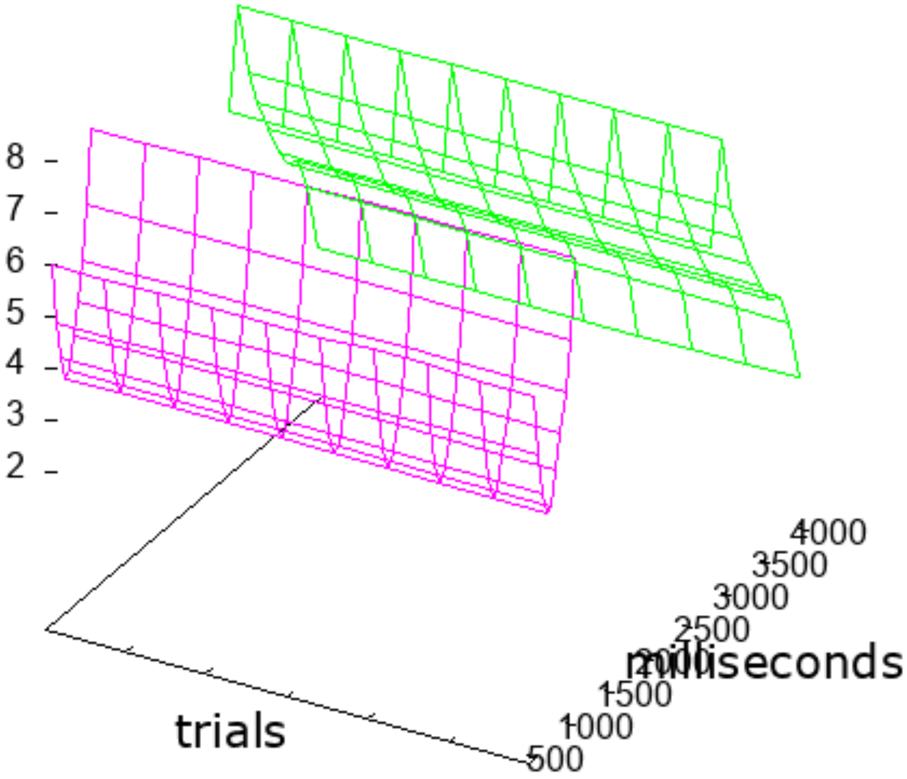
CMSC 337
Algorithms: Design & Practice

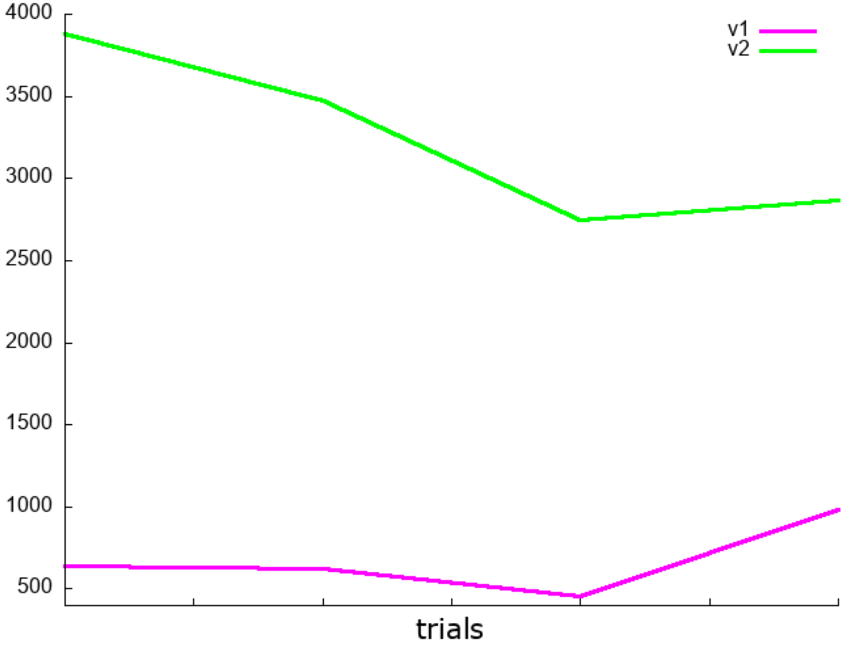
Graphs: Do/Don't

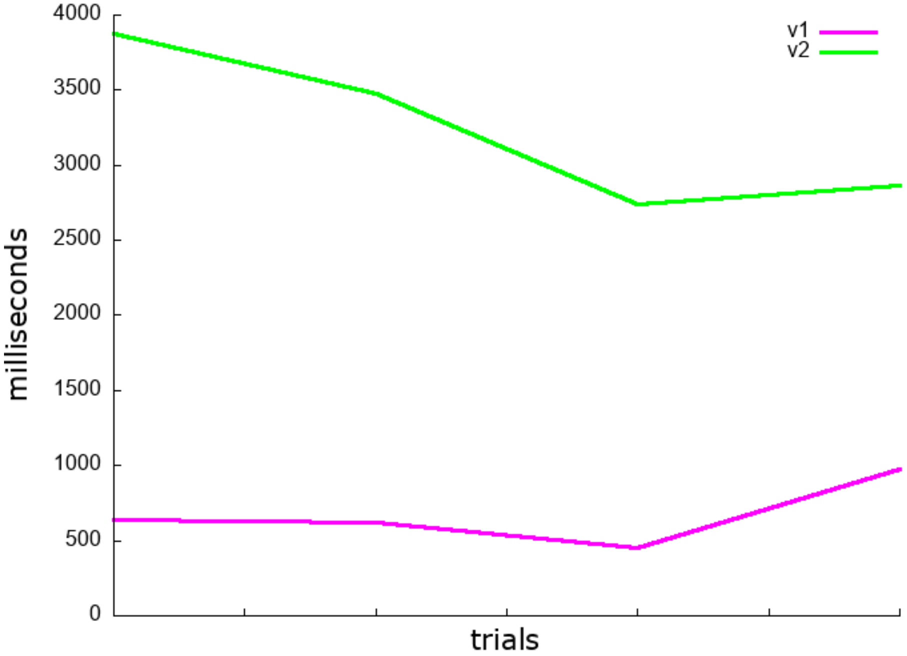
- Simple
- Lies
- Labels
- Appropriate type


8
7
6
5
4
3
2
trials
milliseconds
500
1000
1500
2000
2500
3000
3500
4000

Graphs: Do/Don't

- Simple
- Lies
- Labels
- Appropriate type

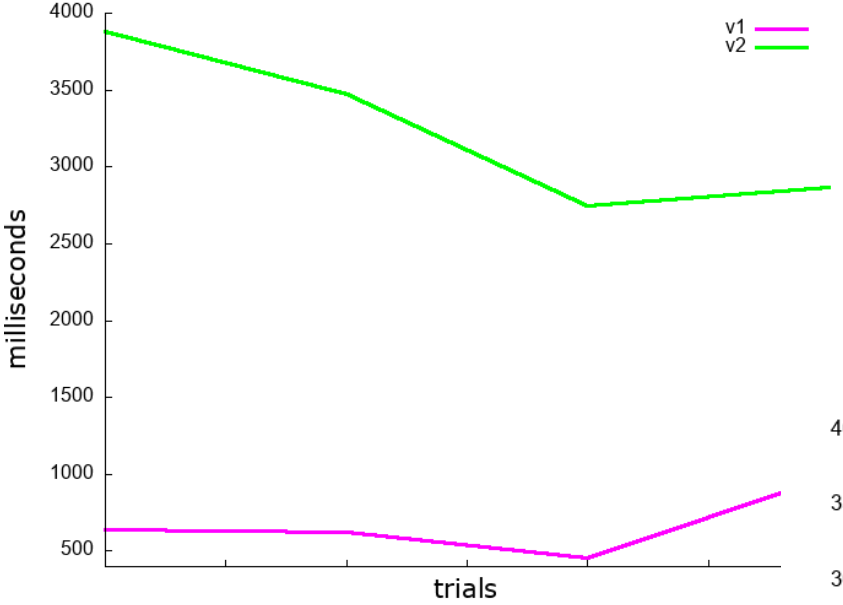

4000
3500
3000
2500
2000
1500
1000
500
v1
v2
trials


4000
3500
3000
2500
2000
1500
1000
500
0
v1
v2
milliseconds
trials

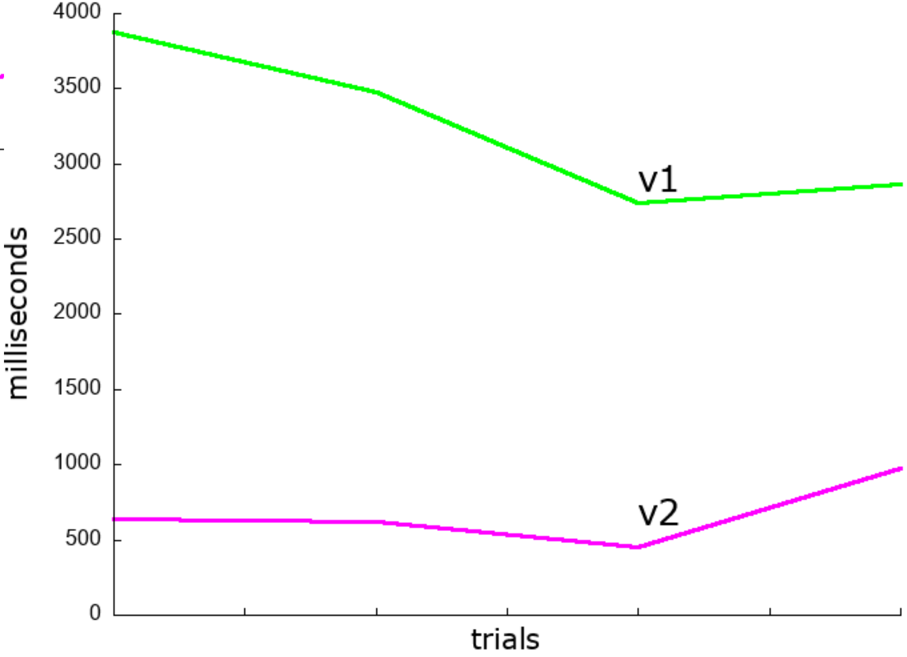
Graphs: Do/Don't

- Simple
- Lies
- Labels
- Appropriate type
- Beautiful

Use labels rather than keys

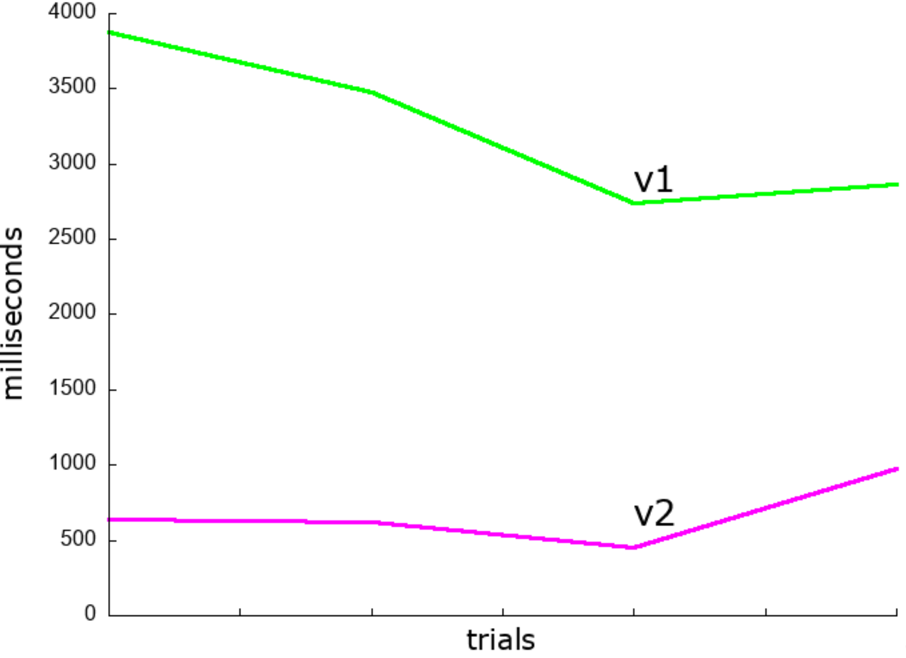


Also, every axis should be labelled

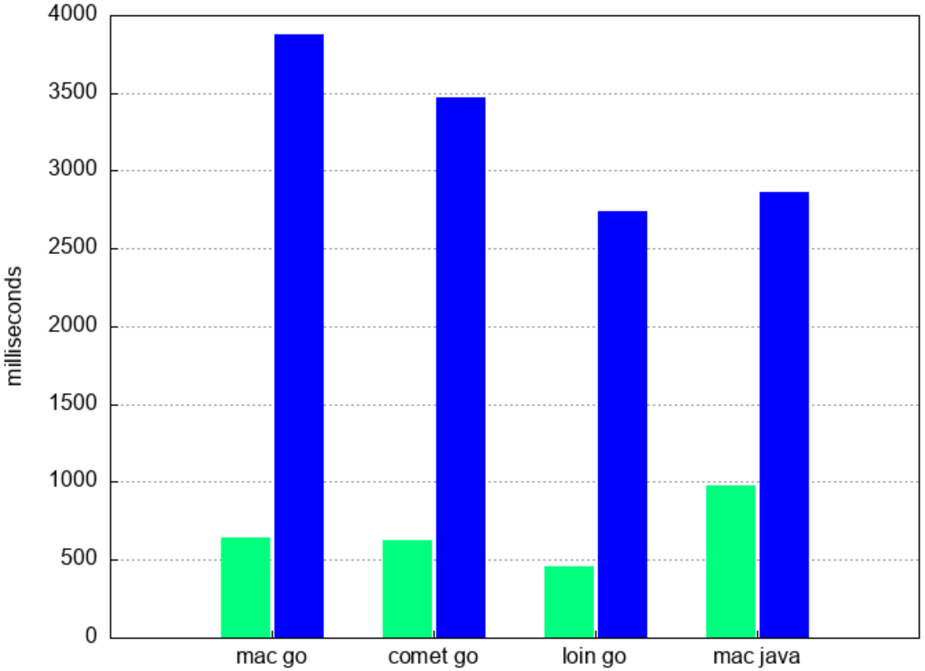


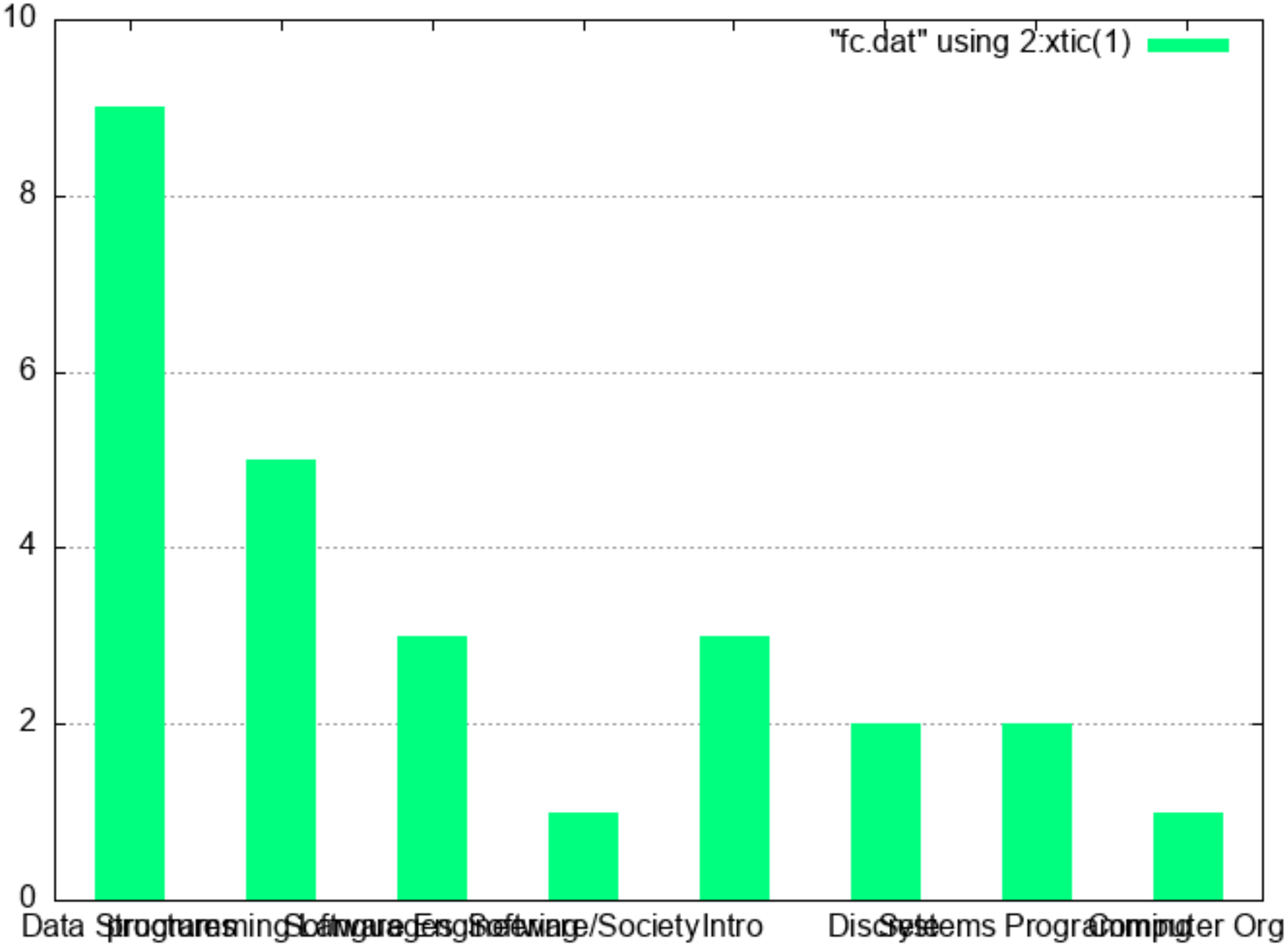
Graphs: Do/Don't

- Simple
- Lies
- Labels
- Appropriate type

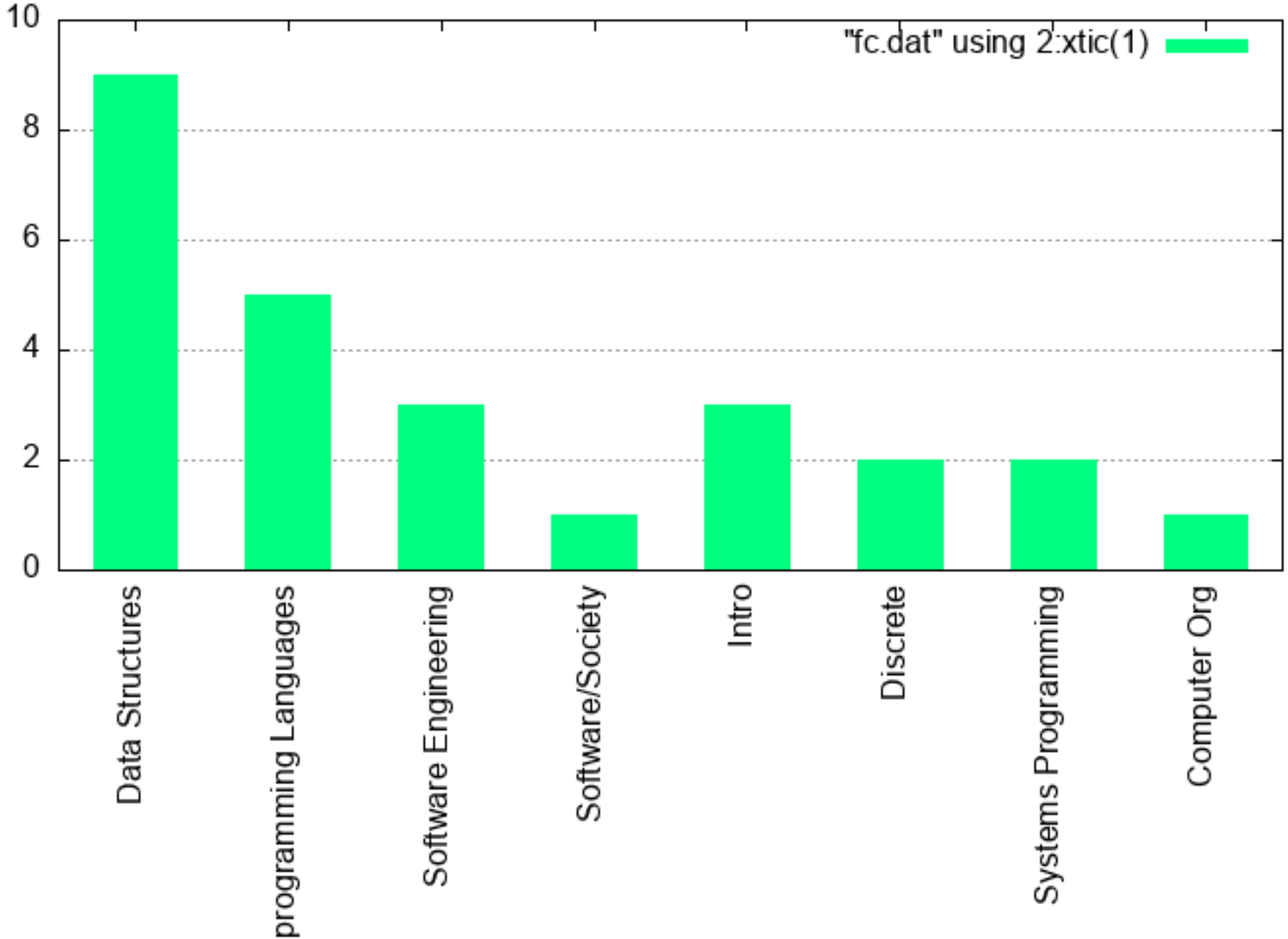


Use the appropriate type

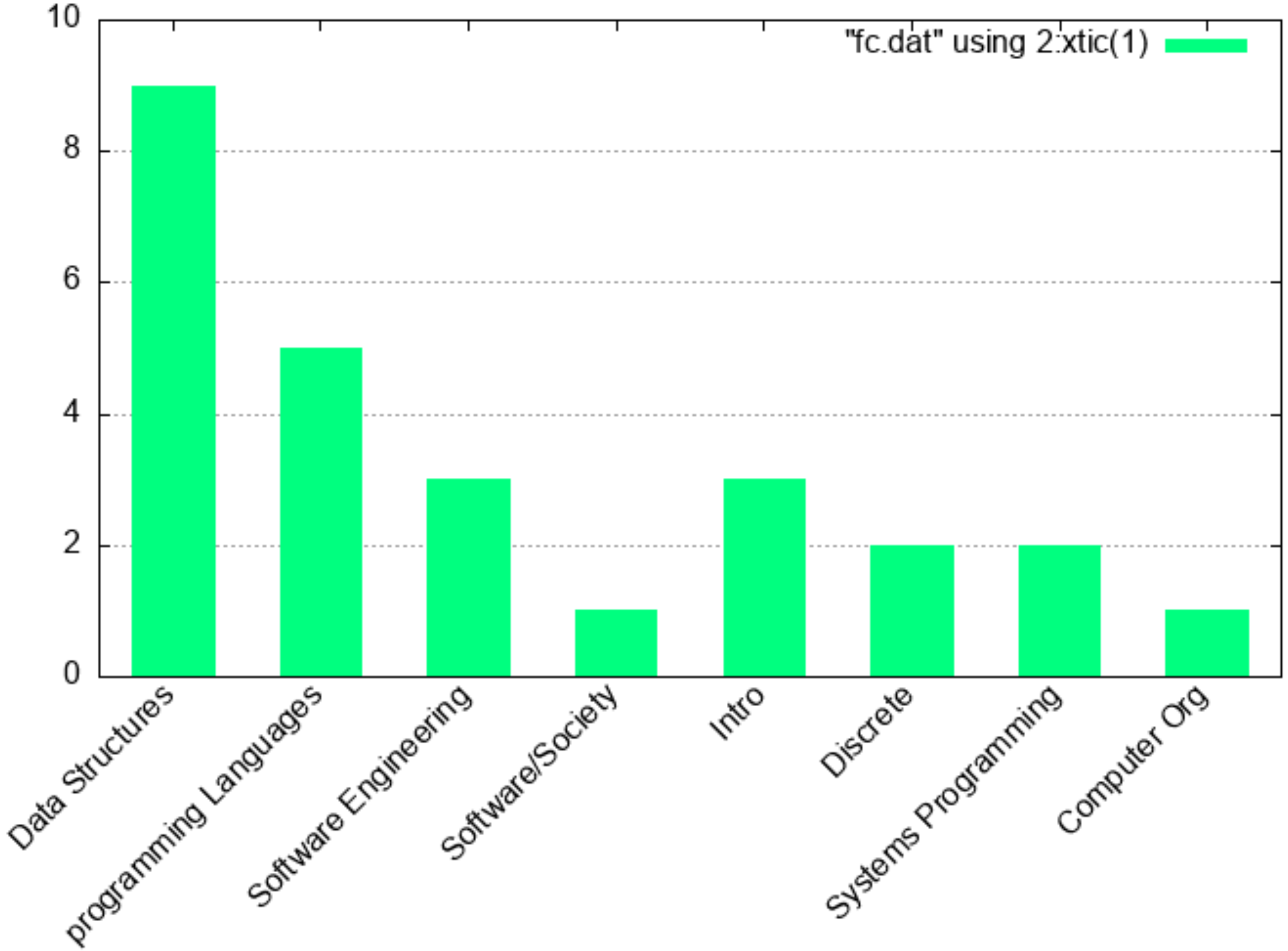


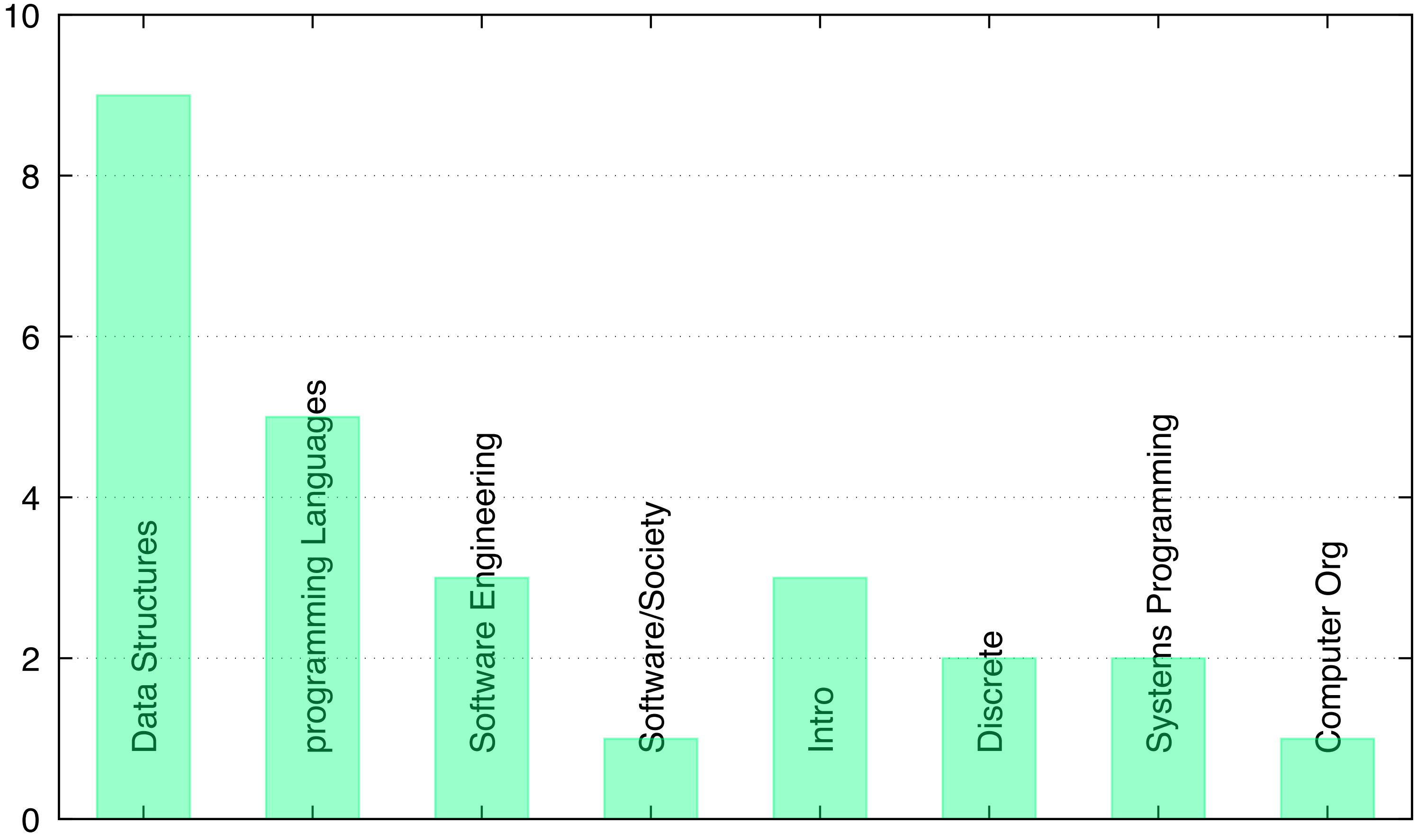


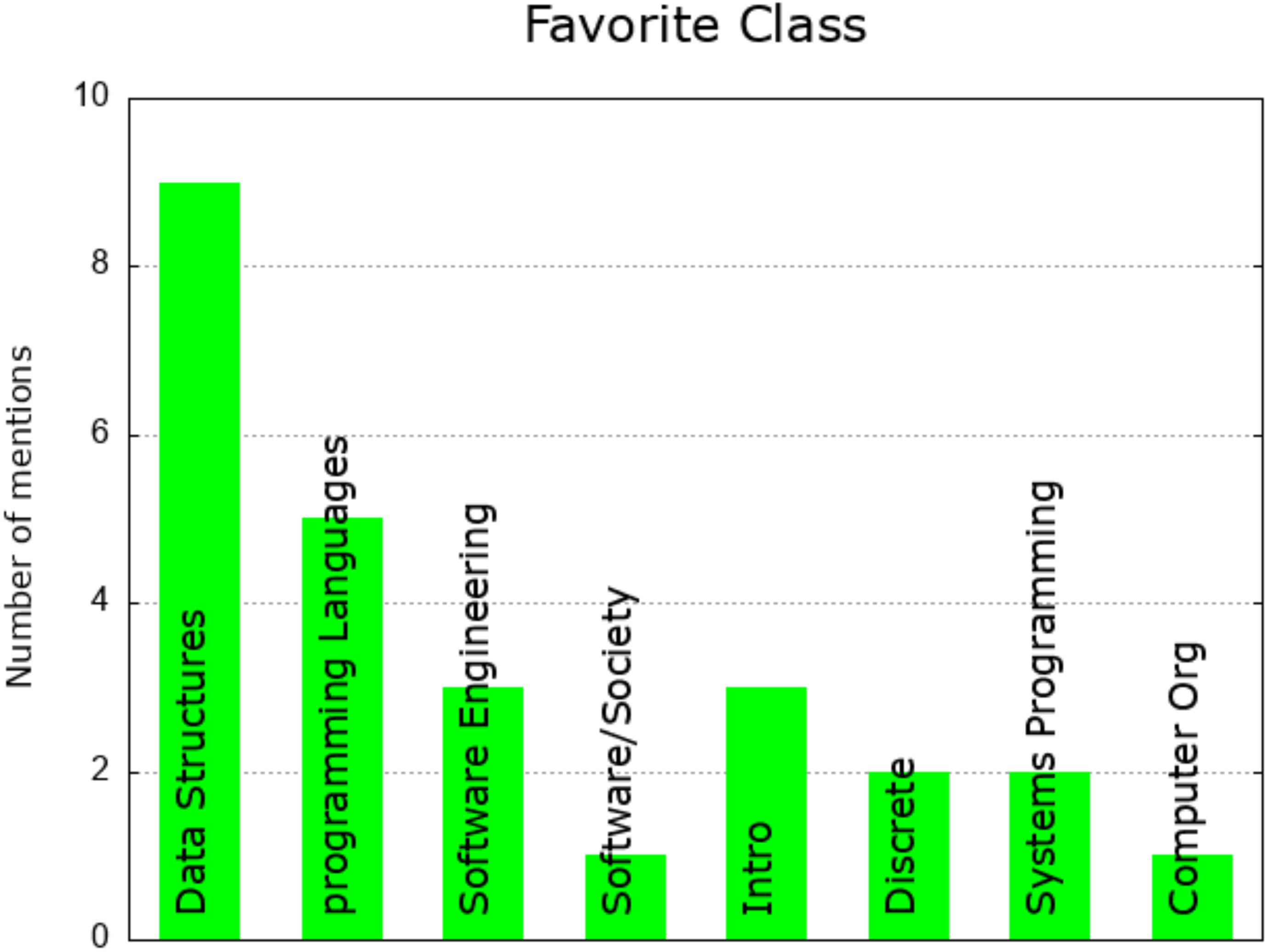
10
8
6
4
2
0
"fc.dat" using 2:xtic(1)
Data Structures
Programming Languages
Software Engineering
Software/Society
Intro
Discrete
Systems Programming
Computer Org



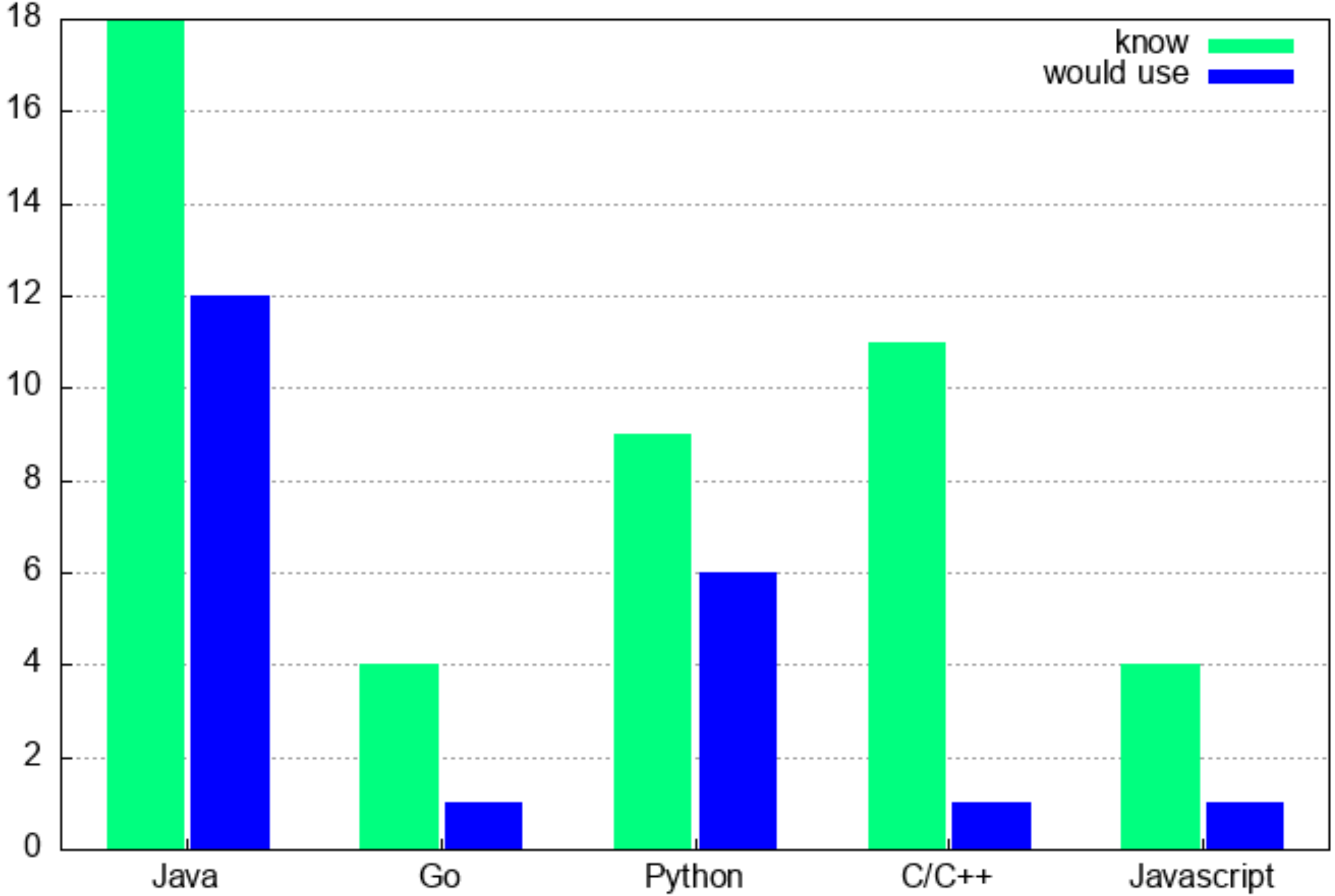
"fc.dat" using 2:xtic(1) 🟩
10
8
6
4
2
0
Data Structures
programming Languages
Software Engineering
Software/Society
Intro
Discrete
Systems Programming
Computer Org


"fc.dat" using 2:xtic(1)
10
8
6
4
2
0
Data Structures
programming Languages
Software Engineering
Software/Society
Intro
Discrete
Systems Programming
Computer Org

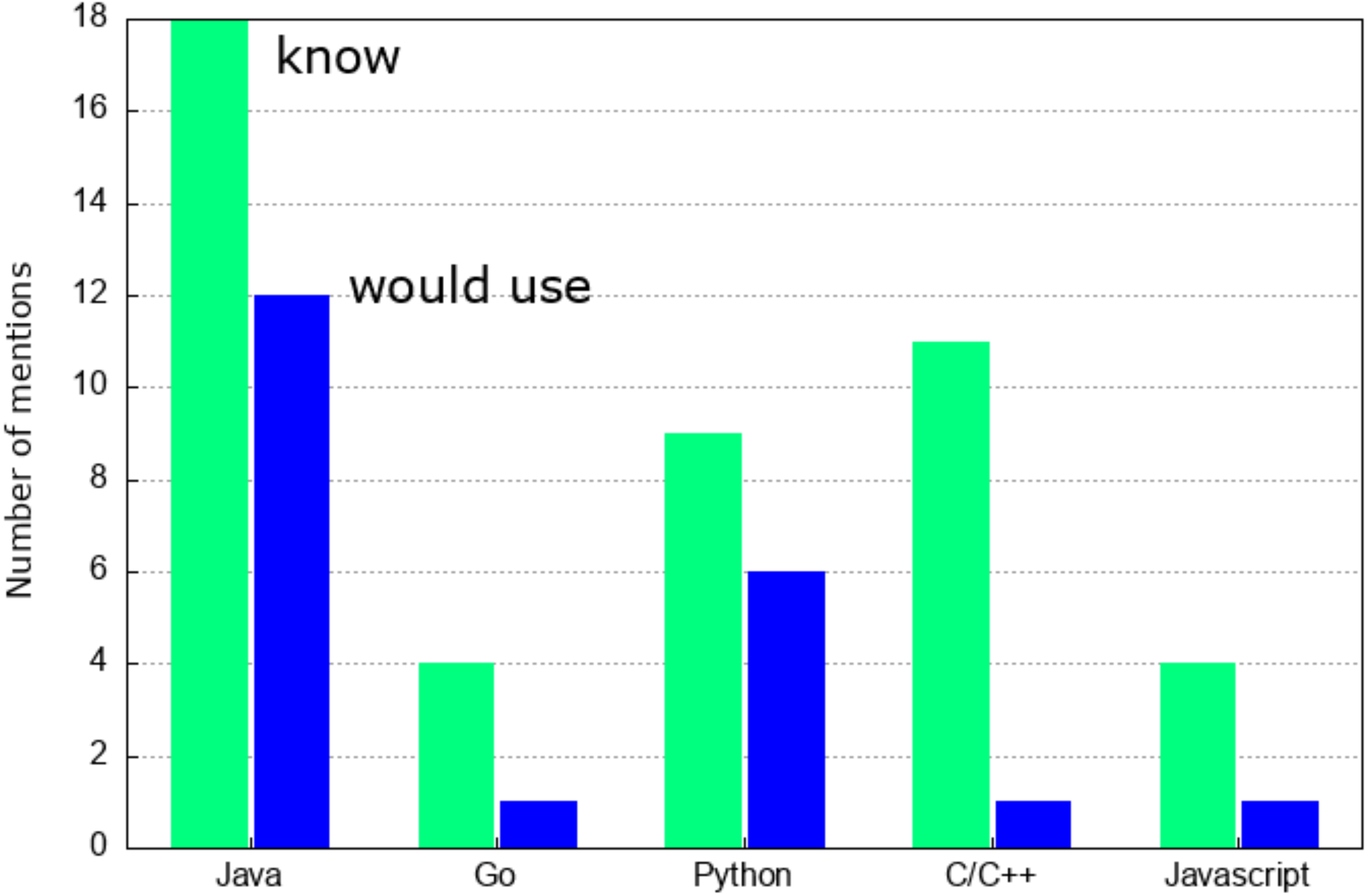

10
8
6
4
2
0
Data Structures
programming Languages
Software Engineering
Software/Society
Intro
Discrete
Systems Programming
Computer Org


Favorite Class
Number of mentions
0
2
4
6
8
10
Data Structures
programming Languages
Software Engineering
Software/Society
Intro
Discrete
Systems Programming
Computer Org

Programming Language



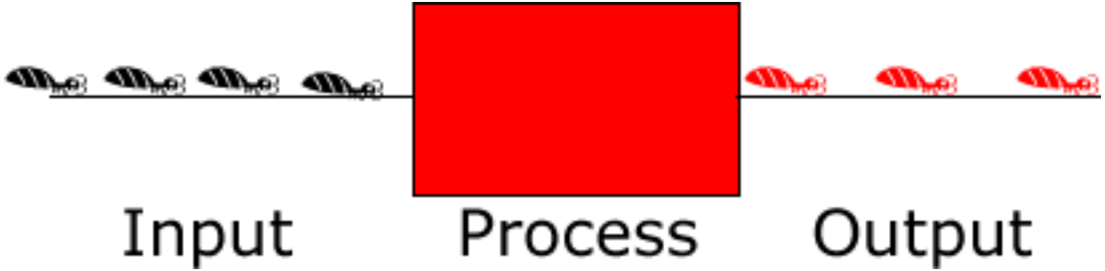
Programming Language



Algorithms

A Bird’s Eye View

Information Processing



Problem Solving
(Investigation of Processes)

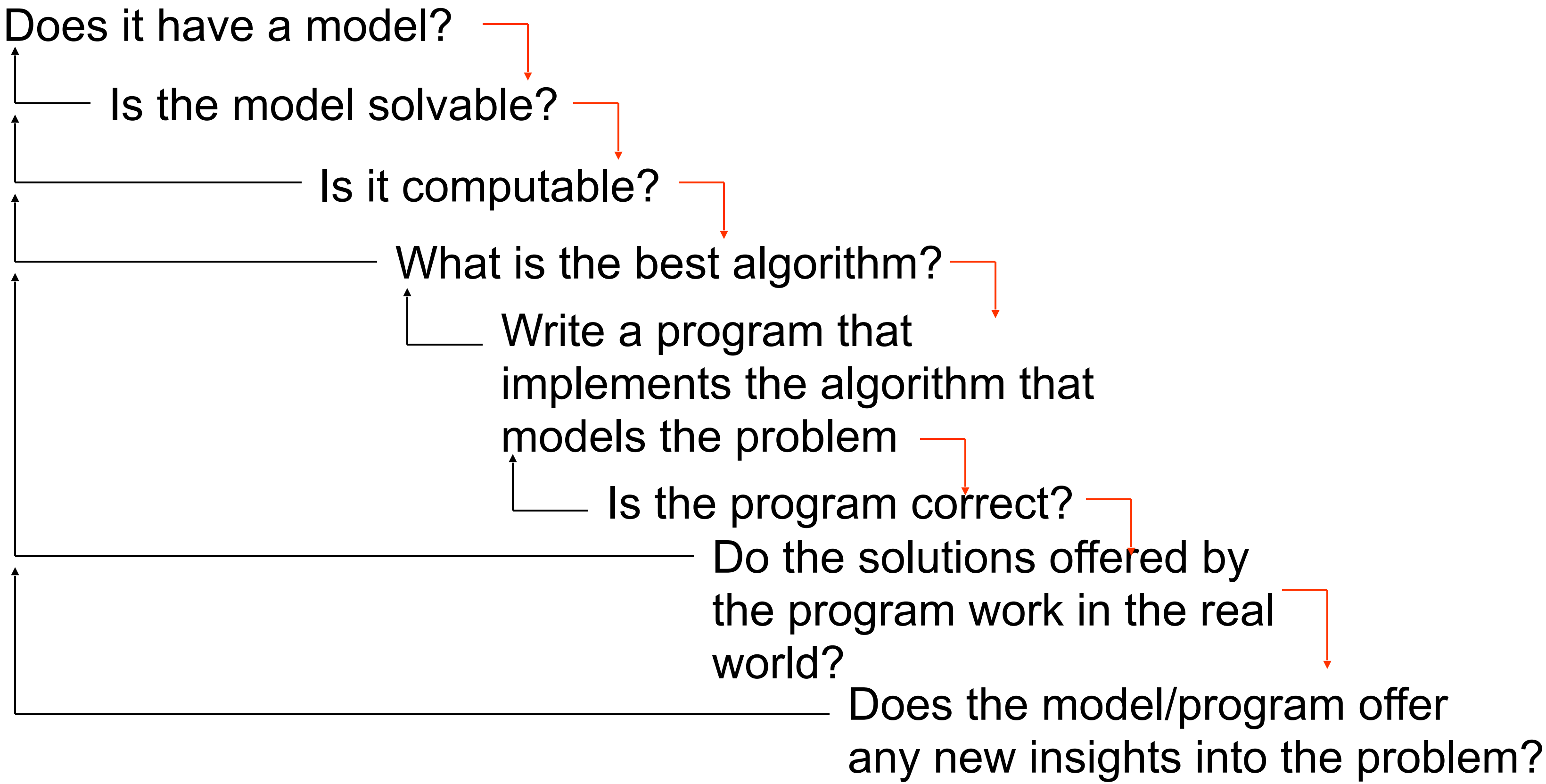
- Given a dictionary of english words, what are all the anagram classes? (e.g. earthling, haltering, lathering)
- Given the details of a tropical depression, can you predict if it will become a hurricane? What path will it take?
- Can you play the game of chess in a way that guarantees a win or a draw?
- How does the mind work?
- What is the most optimal way to get from here to there?
- What is the square root of 42?
- What impact will the Fed raising short term interest rates have on international currencies?
- What is the meaning of life?

Investigation: Asking Questions

Given: A problem/process.

- Does it have a model?
- Is the model solvable?
- Is it computable?
- What is the best algorithm for it?
- Write a computer program that implements the algorithm.
- Is the program equivalent to the model?
- Does the model lend any new insights into the problem/process?

Given: A Problem



Does it have a model?

Is the model solvable?

Is it computable?

What is the best algorithm?

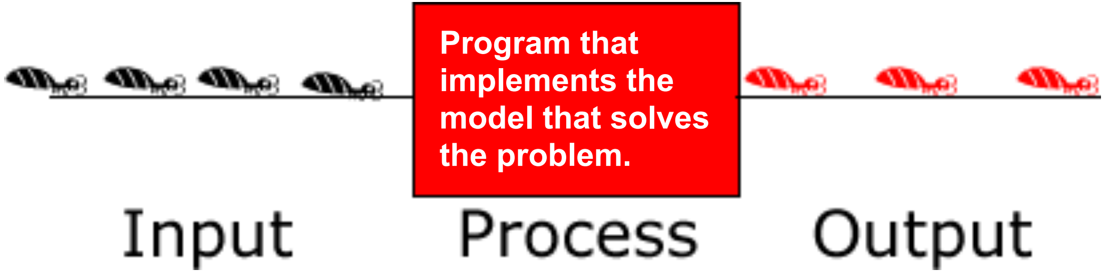
Write a program that implements the algorithm that models the problem

Is the program correct?

Do the solutions offered by the program work in the real world?

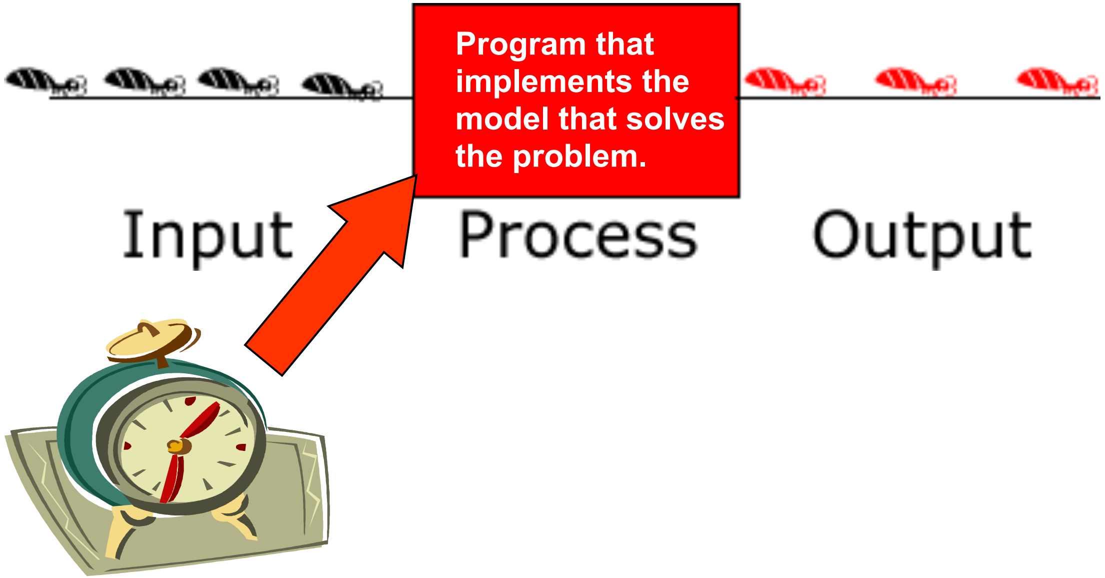
Does the model/program offer any new insights into the problem?

Information Processing



Program that implements the model that solves the problem.

Information Processing



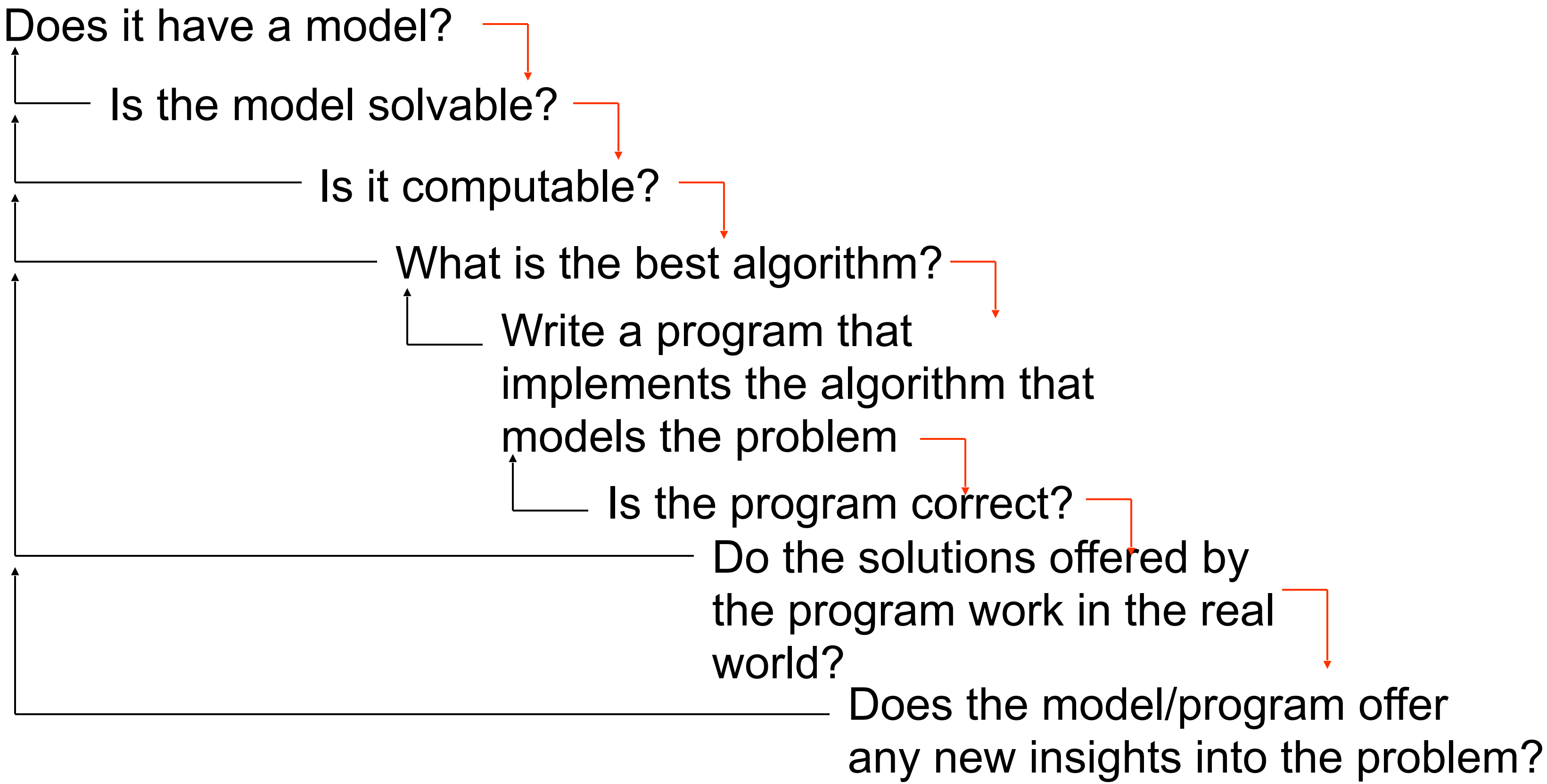
Program that implements the model that solves the problem.

Input

Process

Output

Given: A Problem



Does it have a model?

Is the model solvable?

Is it computable?

What is the best algorithm?

Write a program that implements the algorithm that models the problem

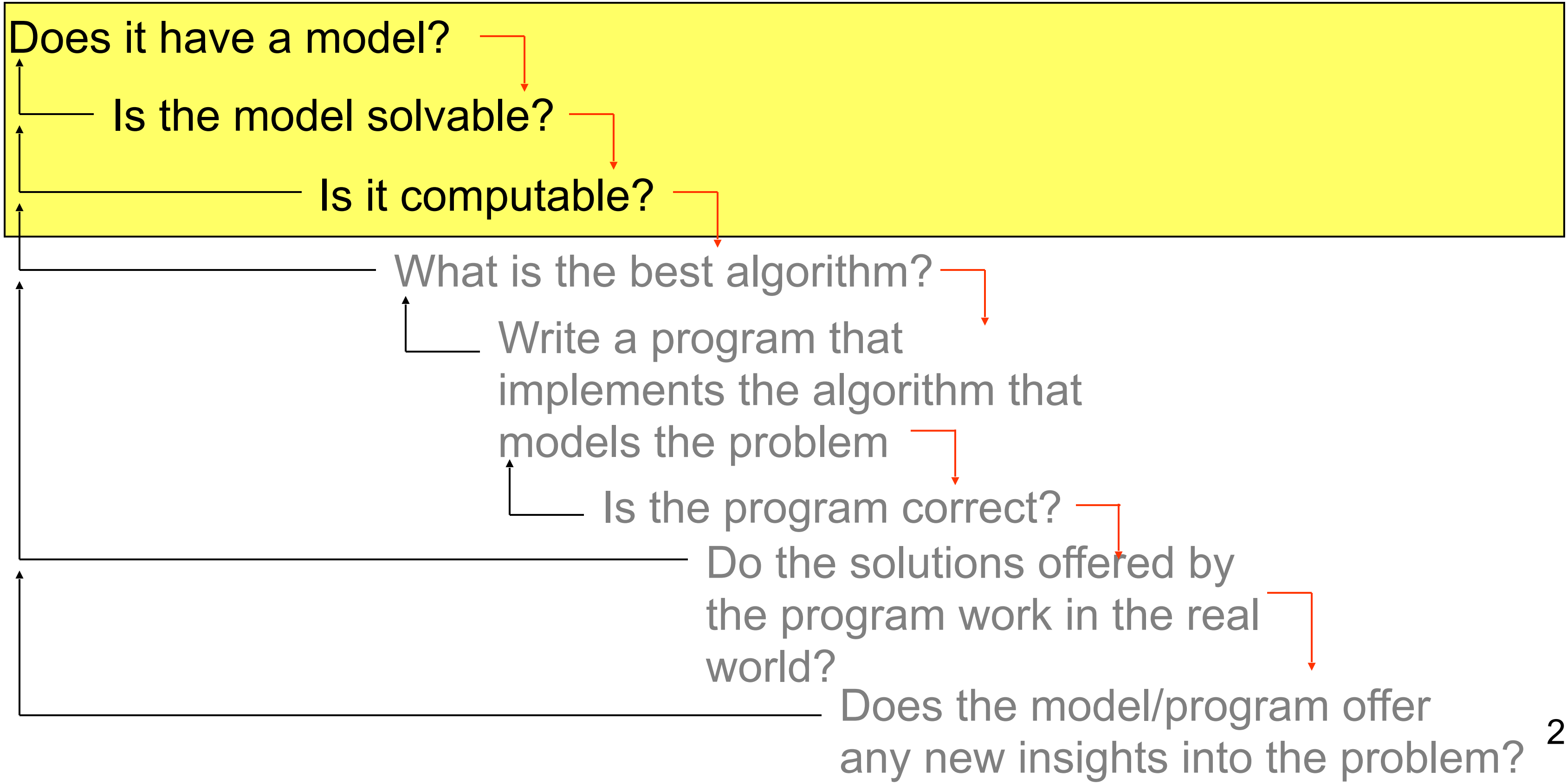
Is the program correct?

Do the solutions offered by the program work in the real world?

Does the model/program offer any new insights into the problem?

Computability:

Problems that can be solved by algorithms (Turing Machines)



Does it have a model?

Is the model solvable?

Is it computable?

What is the best algorithm?

Write a program that implements the algorithm that models the problem

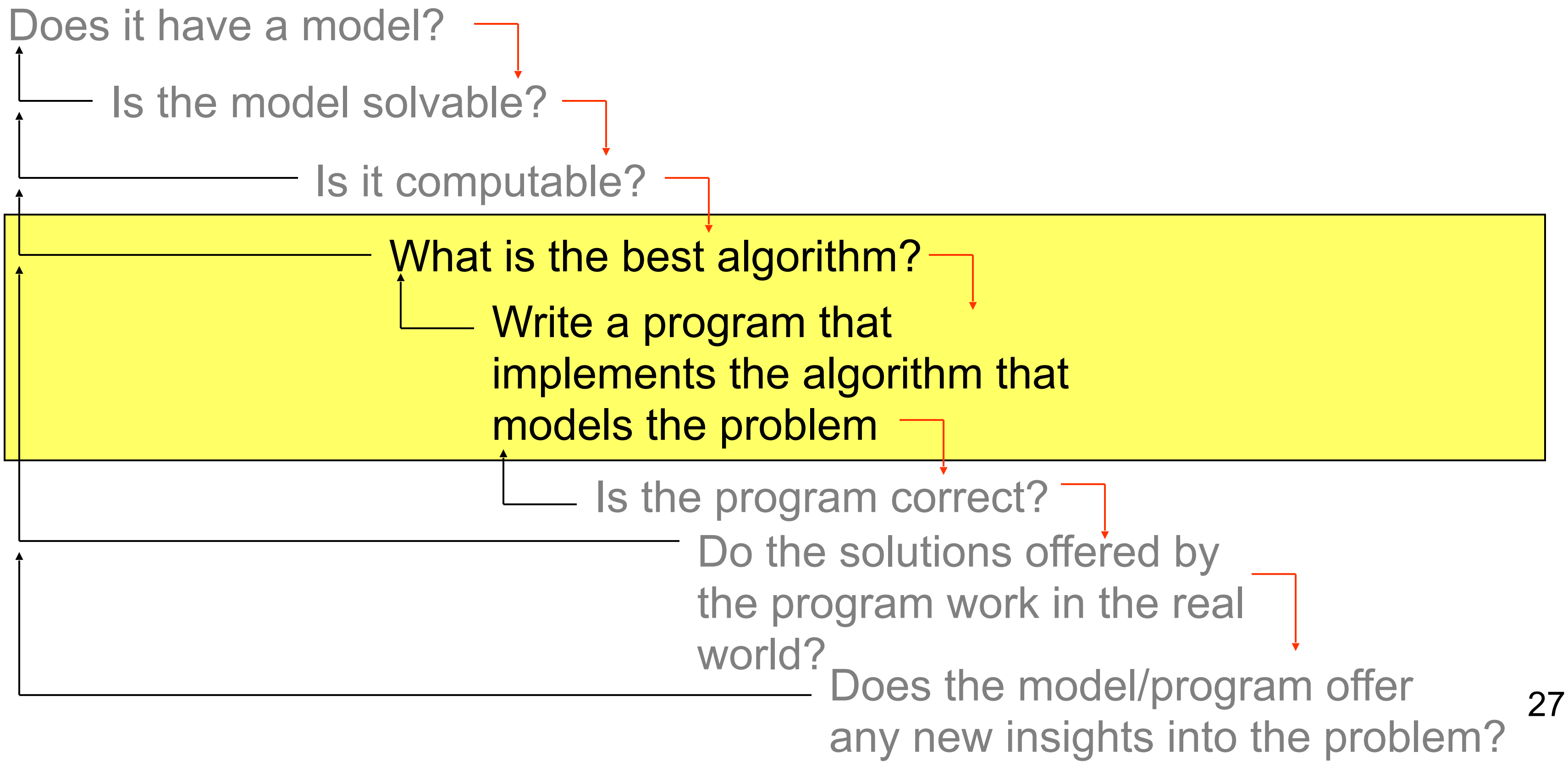
Is the program correct?

Do the solutions offered by the program work in the real world?

Does the model/program offer any new insights into the problem?

Complexity Theory:

Computational Resources required (time & space)



Does it have a model?

Is the model solvable?

Is it computable?

What is the best algorithm?

Write a program that implements the algorithm that models the problem

Is the program correct?

Do the solutions offered by the program work in the real world?

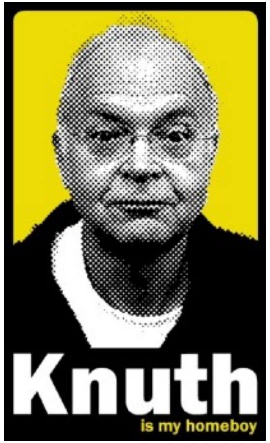
Does the model/program offer any new insights into the problem?

What is an *algorithm*?

A set of instructions arranged in a specific order is a *procedure*.

Similar to a *recipe*, *process*, *method*, *technique*, *procedure*, *routine*, *rigmarole*, except the word "algorithm" connotes just a little something different.

An algorithm is a finite, definite, effective procedure, with some output.



Donald Knuth: The Art of Computer Programming, Volume 1: Fundamental Algorithms, 3rd edition, 1997.

Algorithm (properties)

Finite

There must be an end to it within a reasonable time

Definite

Precisely definable in clearly understood terms, no “pinch of salt” type vagaries, or possible ambiguities

Effective

It must be possible to actually carry out the steps

Procedure

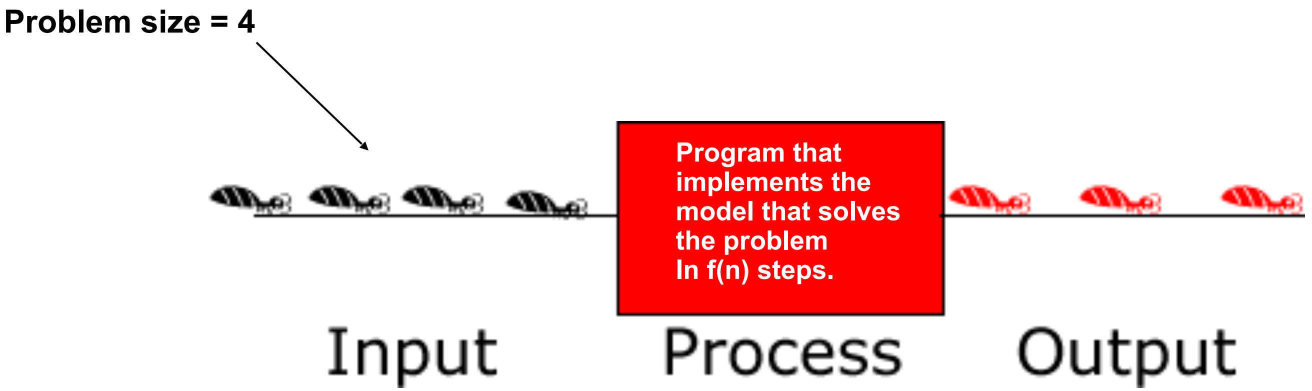
The sequence of specific steps

Output

Unless there is something coming out of the computation, the result will be unknown!

Problem Size

- Time complexity of a problem is the number of steps that it takes to solve an instance of the problem as a function of the size of the input. i.e. if the input is of size, *n*, it will take *f(n)* steps to solve it.


Problem size = 4
Program that implements the model that solves the problem In f(n) steps.
Input
Process
Output

How long to sort 10 million numbers?

Computer A

Speed: 10^{10} instructions/sec
Running O(n^2) sort
Requires $2n^2$ instructions

How long will it take?

Computer B

Speed: 10^7 instructions/sec
Running O($n \log n$) sort
Requires $50\ n \log n$ instructions

How long will it take?

How long to sort 10 million numbers?

Computer A

Speed: 10^{10} instructions/sec
Running O(n^2) sort
Requires $2n^2$ instructions

$$\frac{2 * (10^7)^2}{10^{10}} \approx 20{,}000s$$

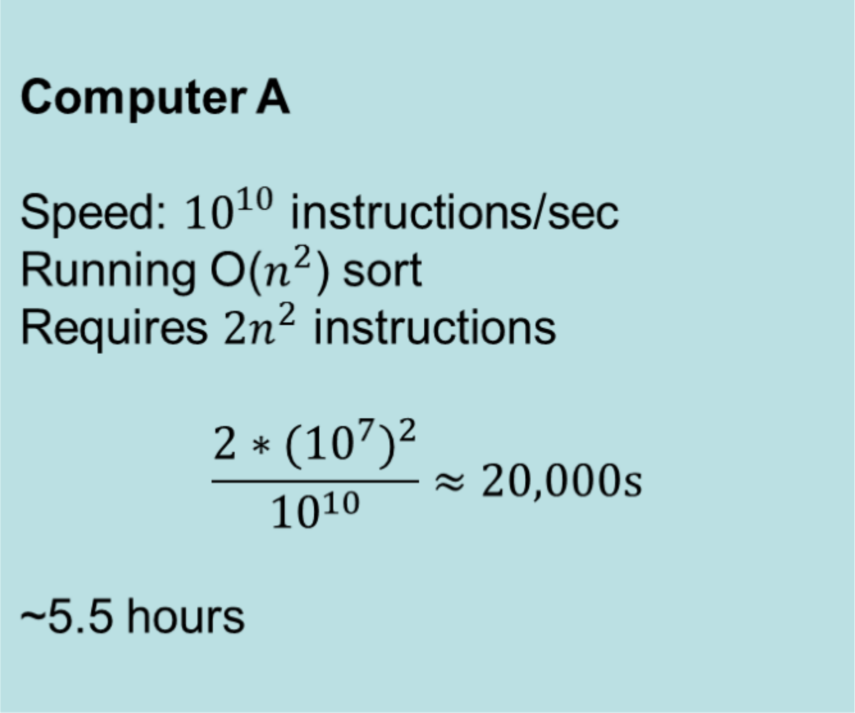
~5.5 hours

Computer B

Speed: 10^7 instructions/sec
Running O($n \log n$) sort
Requires $50\ n \log n$ instructions

How long will it take?

How long to sort 10 million numbers?

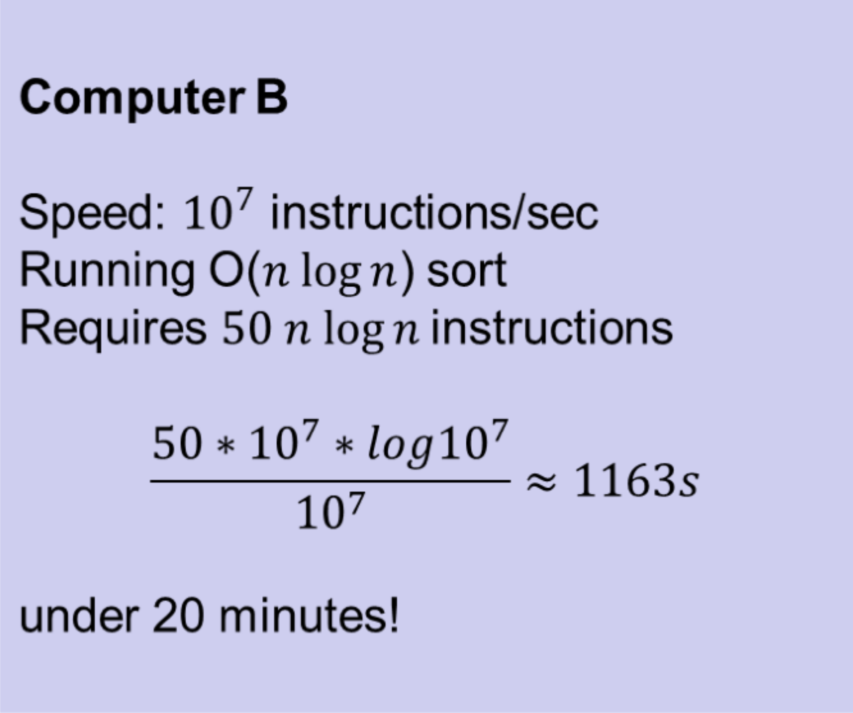


Computer A

Speed: 10^{10} instructions/sec
Running O(n^2) sort
Requires $2n^2$ instructions

$$\frac{2 * (10^7)^2}{10^{10}} \approx 20{,}000s$$

~5.5 hours



Computer B

Speed: 10^7 instructions/sec
Running O($n \log n$) sort
Requires $50\ n \log n$ instructions

$$\frac{50 * 10^7 * log 10^7}{10^7} \approx 1163s$$

under 20 minutes!

How long to sort 10 million numbers?

Computer A

Speed: 10^{10} instructions/sec
Running O(n^2) sort
Requires $2n^2$ instructions

$$\frac{2 * (10^7)^2}{10^{10}} \approx 20{,}000s$$

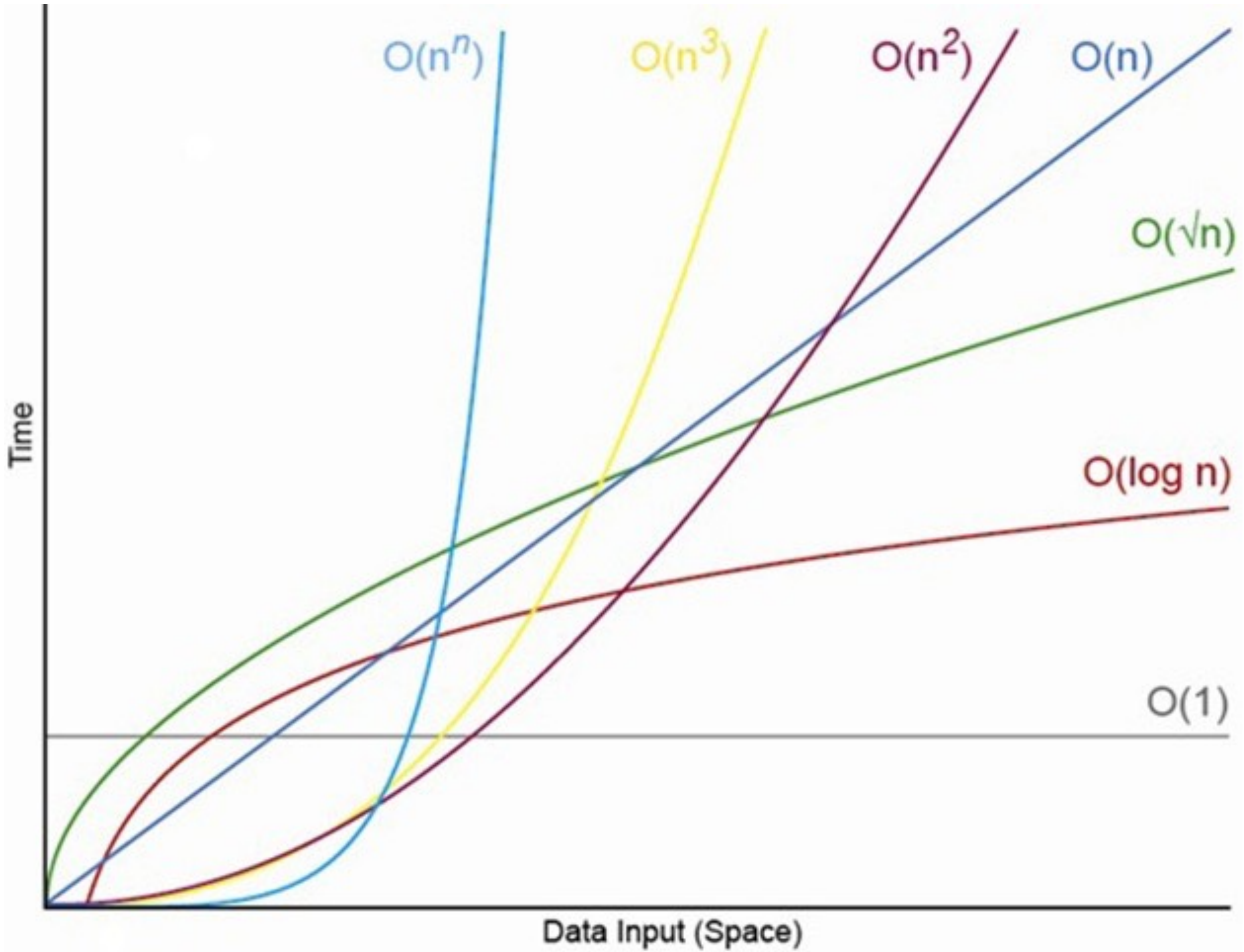
If running $50\; n \log n$ program: < 2s!!

Computer B

Speed: 10^7 instructions/sec
Running O($n \log n$) sort
Requires $50\; n \log n$ instructions

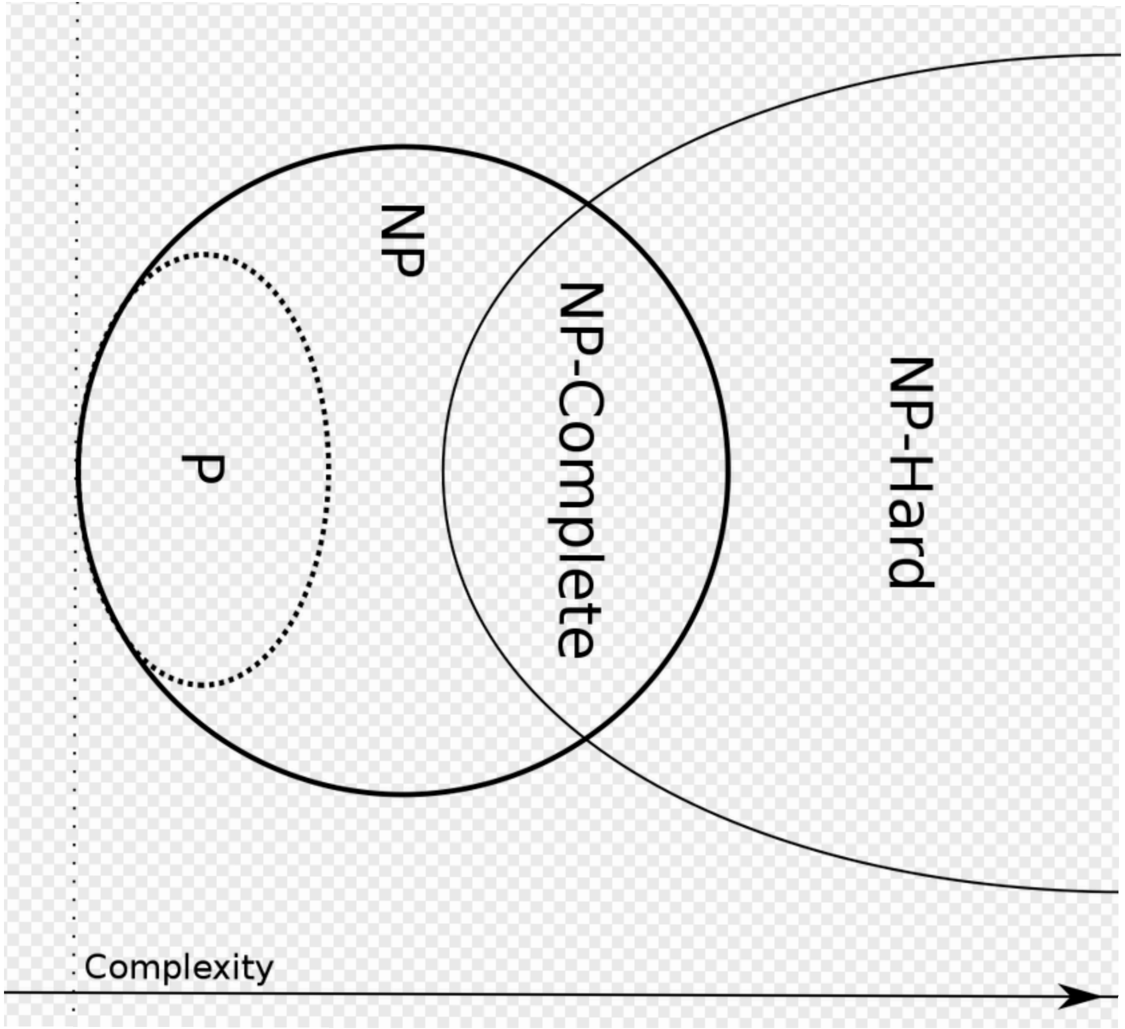
$$\frac{50 * 10^7 * log 10^7}{10^7} \approx 1163s$$

under 20 minutes!



http://apelbaum.wordpress.com/2011/05/

P = NP?

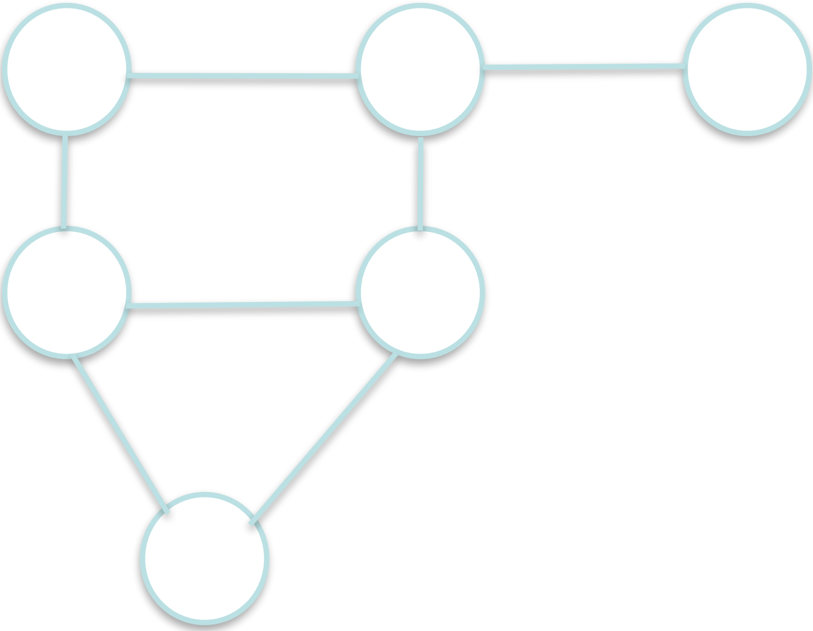


from wikipedia

NP-Complete

- NP = Non-determinitic Polynoimial
- in NP == Solution is verifiable in P time
- problem is provably equivalent to other NP complete problems

- vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge.



Vertex Cover Algorithm

- Find the minimum vertex cover of a graph
  - We will discuss graph representations, just make something up for now

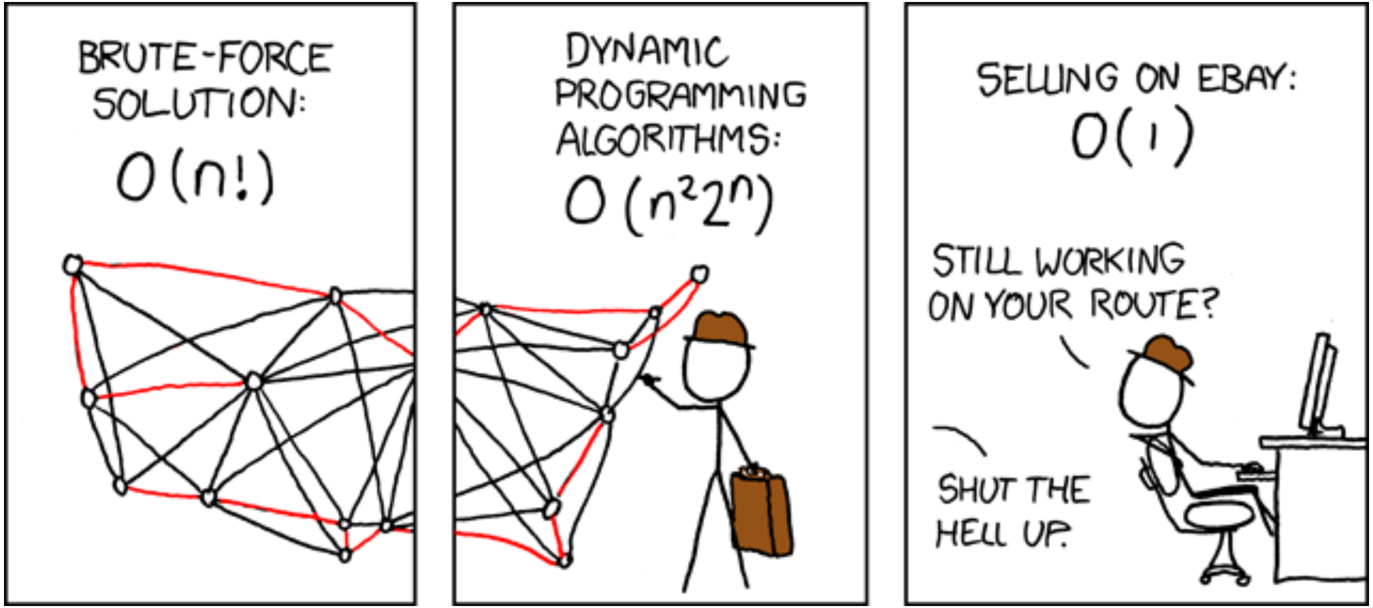
Vertex Cover Algorithm

- Optimal algorithm

- Naive algorithm

- Greedy Algorithm

xkcd??



- More on xkcd.com

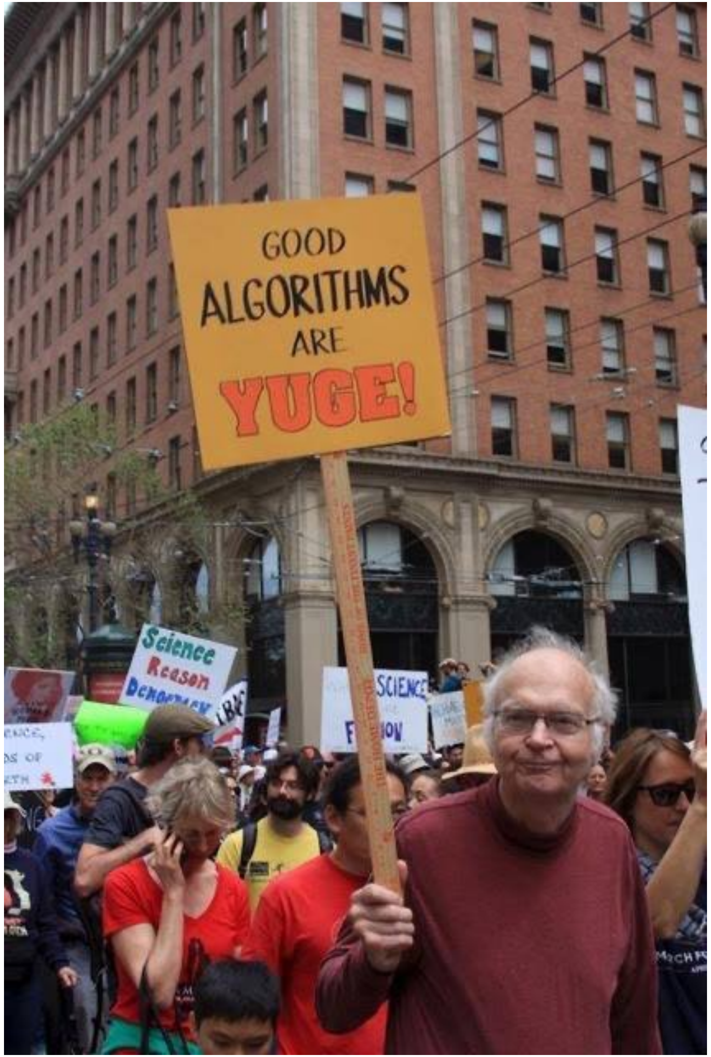
Algorithm for Algorithm Development

```
def algorithmDevelopment(problemSpec):
    correct = false
    while not correct or not fastEnough(runningTime):
        algorithm = deviseAlgorithm(problemSpec)
        correct = analyzeCorrectness(algorithm)
        runningTime = analyzeEfficiency(algorithm)
return algorithm
```

Algorithm for Program Development

```
def programDevelopment(algorithm, testSuite):
    language = pickLanguage(algorithm)
    program = code(algorithm, program)
    do:
        check = false
        while not check:
            program = debug(program)
            check = verifyProgram(program, testSuite)

        performance = measure(performance)
    while not acceptable(performance)
```


GOOD
ALGORITHMS
ARE
YUGE!
Science
Reason
Democracy
SCIENCE

An algorithm to consider

- Given two lists of integers
  - call these A and B
- Find: min(abs(A[i]-B[j])