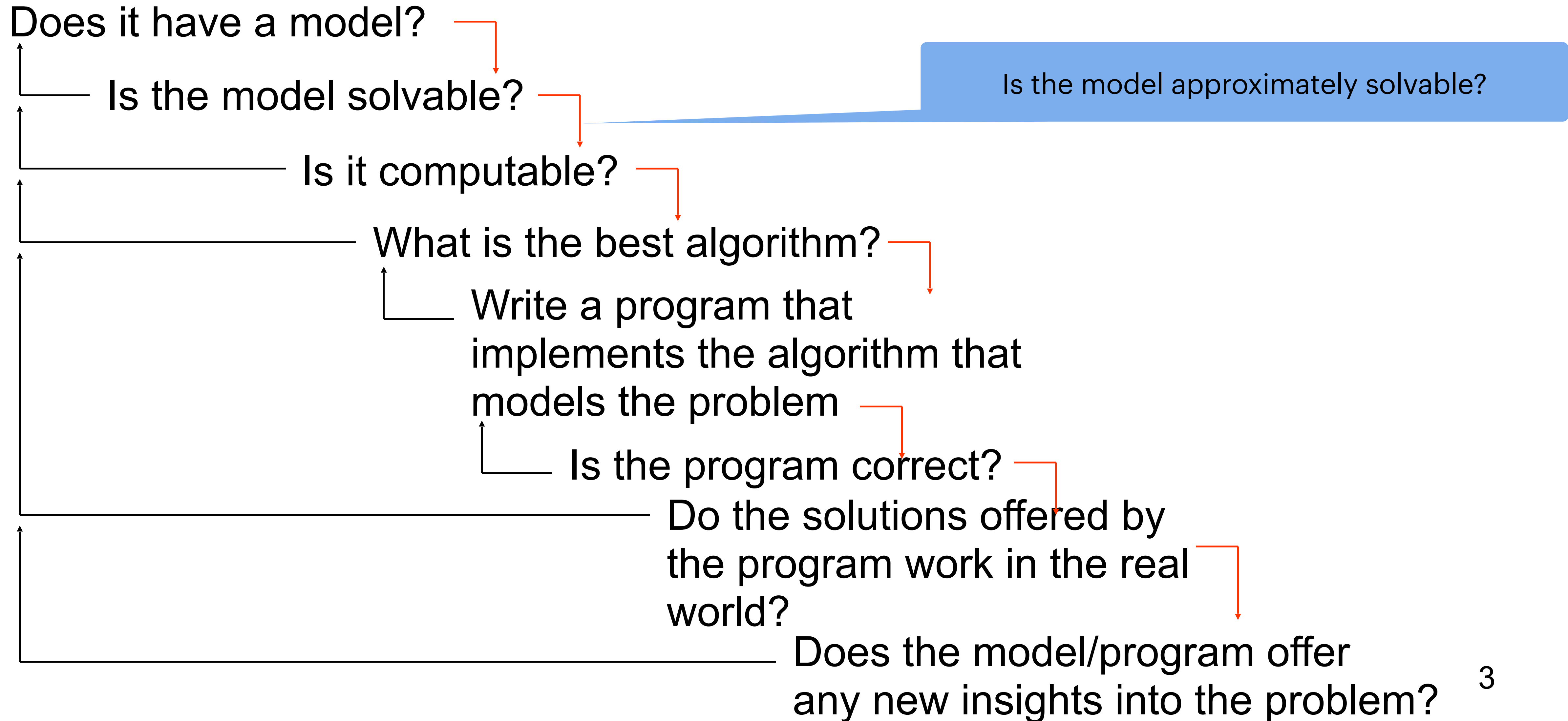# Thinking about Algorithms

## and lots of them

Feb 19

1. Dynamic Programming

2. Binary Search

3. Depth-first search

4. Breadth-first search

5. Dijkstra's algorithm

6. Greedy Algorithm

7. Sorting and Searching

8. Backtracking

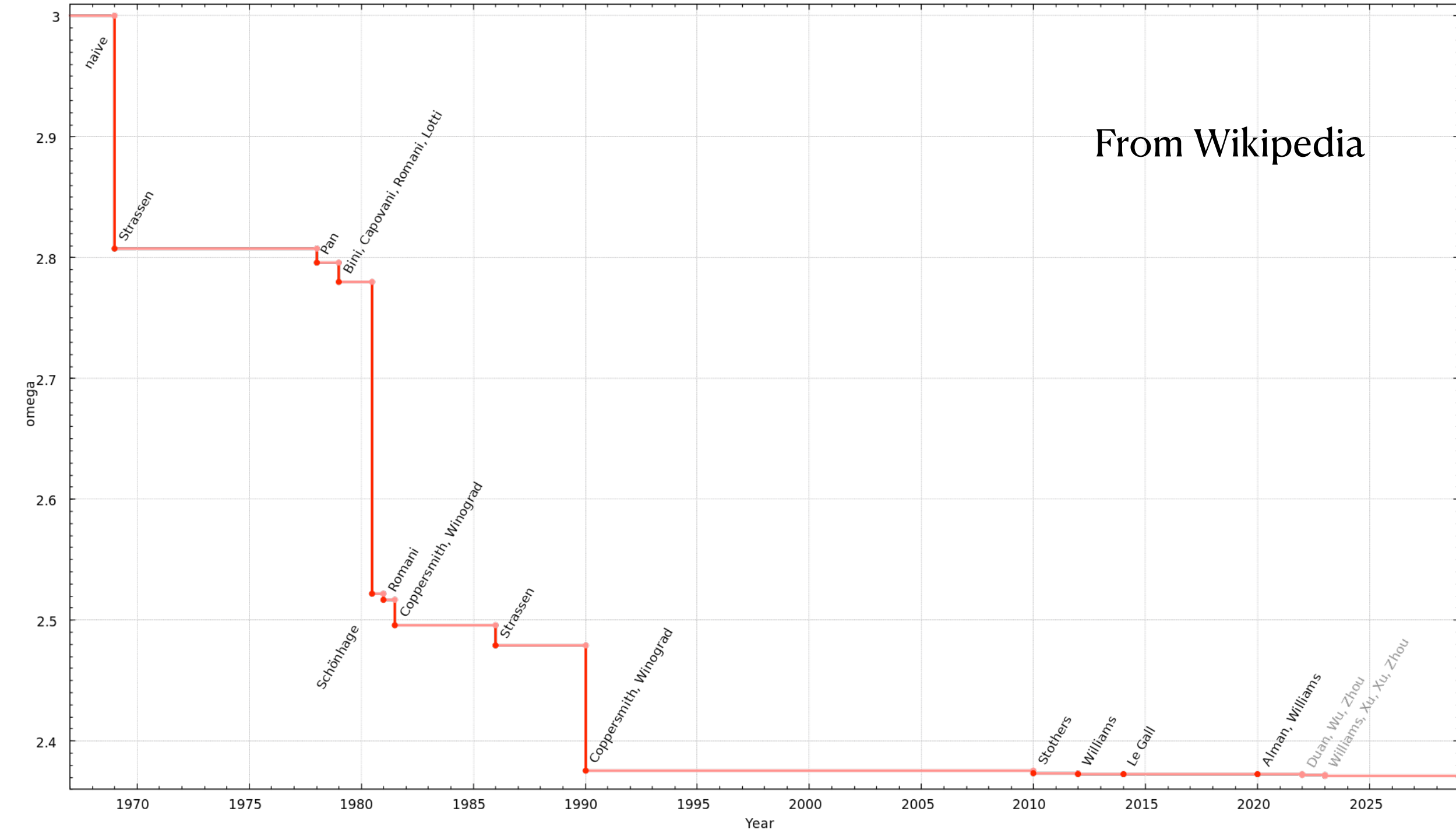9. Bit Manipulation

10. Divide and Conquer

# Given: A Problem

Does it have a model?

Is the model solvable?

Is the model approximately 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?

3

# Multiplying Matrices

- Give an algorithm for multiplying an NxN matrix with another NxN matrix

- Time Complexity?

  - 

From Wikipedia

# Anagrams

- Given a list of words design a program / data structure that can efficiently answer the question:

  - Is a new word an anagram of any word in the original list

- What is the time complexity of your algorithm

  - preprocessing the list

  - answering yes/no about a new word

# Anagrams -- part 2

- Alternate Question: Given a list of words, find all words that are anagrams of each other

  - Can you use this to help with the first anagram task?


- Time complexity?

# Closest pair of integers

- given two lists of positive integers, what is the closest pair?

  - formally: given sets of positive integers X and Y find indices i and j such that abs(X[i]-Y[j]) is minimized for all i, j

- Does the algorithm work for all integers? Does it work for real numbers?

- Time Complexity?

# Speed Dating

- Given 2 groups X and Y

  - design an efficient algorithm in which each member of group X meets each member of group Y

- What is time complexity?

# Speed Friending

**"gay speed dating"**

- Given a group X design an efficient algorithm in which each member of the group meets each other member of the group.

- Time Complexity

# Max Pairwise Product

- Given a set of positive integers find the pair of numbers a[i], a[j] such that i!=j and a[i]*a[j] is maximum.

- Suppose rather than pairwise you need N-wise.

- Suppose the numbers could be negative?

- Time complexity

# Last Fibonacci digit

- Give an algorithm for computing the rightmost digit of the Nth fibonacci number. N may be arbitrarily large.

    - So computing entire fibonacci number is not possible

- Time Complexity?

- Space Complexity?

- Can your technique be applied to calculating the last N fibonacci digits

# Collecting Signatures

- You are tasked with collecting the signatures for every tenant of an apartment building.
  - You have a complete list of tenants
    - You know each tenant's schedule
    - To get their signature you merely have to say "hello"
    - You can say hello to an infinite number of people simultaneously
  - You cannot stay at the building.
  - You want to minimize your visits to the building.
  - How many visits will you need? When?
    - How should you structure the tenant's schedule data

# Computability

**Question: Will a program crash?**

- Can anyone write a program that takes some other program as input and simply answer the following yes/no question

  - Will the input program ever crash?

  - This is a variant on the Halting Problem (Turing)

- This is largely taken from MacCormack(2012) ch 10

# Proof by Contradiction

- Suppose there exists a program "mayCrash" that will accepts some inputs then and after processing the inputs it does one of three things:

  - output "YES"

  - output "NO"

  - crash

- Suppose there exists a program "canCrash" that takes as input a program (like mayCrash) and a set of inputs for the input program and outputs

  - YES if the program could crash

  - NO otherwise

# CanCrashMod

- CanCrashMod is identical to canCrash BUT

  - outputs

    - rather that saying yes, it crashes

    - NO otherwise

- Modify CanCrashMod to SelfCanCrashMod

  - crashes when given itself and inputs that would cause CanCrashMod to crash

  - No otherwise

    - Even this is pretty much impossible.   You need a program that is capable of running itself in simulation. Which means that you need the program to have as a part of itself a simulator that can run itself.  ….

      - Can you write a compiler that compiles itself?
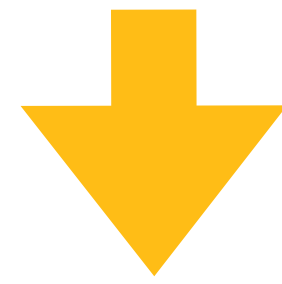
# AntiSelfCanCrashMod

- The negative of SelfCanCrashMod

  - if input would cause a crash when run on itself,  return YES
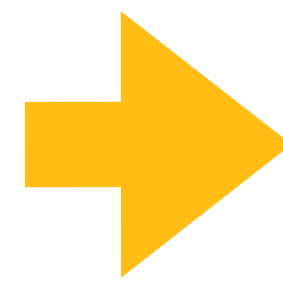
  - crash

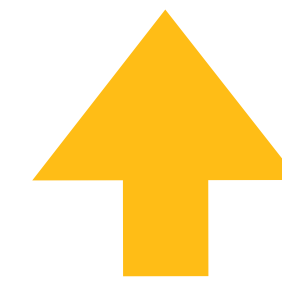**CanCrash**

Outputs:
  YES
  No

**AntiSelfCanCrashMod**

Outputs:
  Yes
  CRASH

**CanCrashMod**

Outputs:
  CRASH
  No

**SelfCanCrashMod**

Outputs:
  CRASH
  No

# Contradiction

**Danger, Will Robinson**
**https://www.google.com/search?client=firefox-b-1-d&q=danger+will+robinson#fpstate=ive&vld=cid:06d64c16,vid:OWwOJlOl1nU,st:0**

- The YES statement of AntiCanCrashSelfMod contradictory!!!

  - program cannot output YES if it has crashed.

- Therefore such a program cannot exist

- QED