Theoretical Computer Science

CSCI 10 - Santa Clara University - Fall 2016
Michael J. Bannister

Announcements

- Resume writing tips
  Tomorrow 5:30pm in Daly Sci 207
- AWM Math Murder Mystery Night
  Tonight 5:30pm in the Sussman Room
- Homework and Report due Friday
- Extra office hours today 1:15pm-3:00pm

Runtime

Question: How do we define the runtime of a program independently of the hardware on which it is run?

Solution: We compute the maximum number of operations the program will perform in a single run. Operations: read, write, arithmetic, logical

Runtime Example: linear search in an unsorted array

```
int array_find(int array[], int sz, int value)
{
    for (int i = 0; i < sz; i++)
    {
        if (array[i] == value) return i;
    }
    return sz;
}
```

Runtime Example: binary search in a sorted array

```
int sorted_array_find(int array[], int sz, int value)
{
    int front = 0;
    int back = sz - 1;
    while (front <= back)
    {
        int k = (front + back) / 2;
        if (array[k] == value) {
            return k;
        } else if (array[k] < value) {
            front = k + 1;
        } else { // array[k] > value
            back = k - 1;
        }
    }
    return sz;
}
```

Problems vs. Programs

Problem: Search for a value in a unsorted array

Program: linear search

Problem: Search for a value in a sorted array

Program: linear search or binary search

Given a problem we want the fastest program solving it!

Complexity Classes

The goal is to classify problems according to their inherent computational difficulty, and then discover relationship between these classes.

Searching for a value in an unsorted array is a harder problem than searching for a value in a sorted array.

P
Polynomial Time

Consists of problems whose runtime is $< an^b + c$ for some fixed constants a, b, c.

Examples:

- Finding a value in an array
- Greatest common divisor
- Sorting
- Determining if a number is prime
- Matrix multiplication

NP
Nondeterministic Polynomial

Consists of problems whose answer can be check in polynomial time.

Examples:

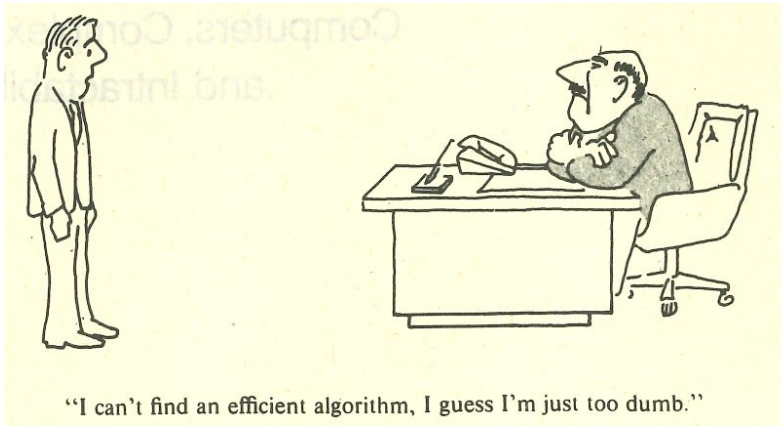
- Graph coloring
- Traveling salesperson problem
- Knapsack Problem

A Million Dollar Question
P = **NP**?

If an answer is easy to check, was the question easy to answer?

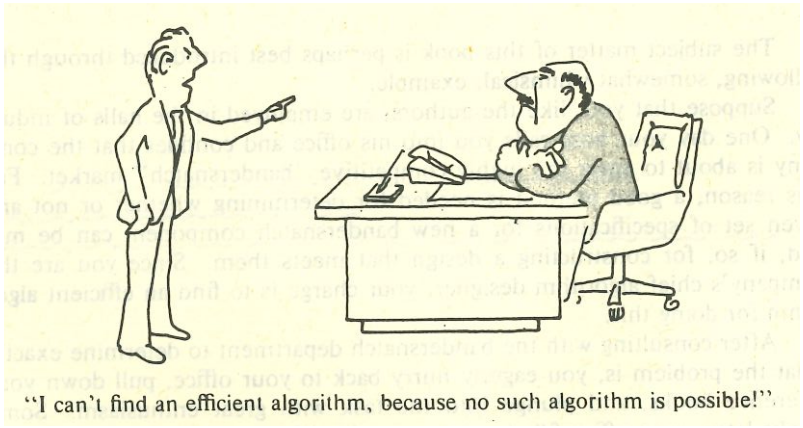
What we know:

- Identified some of the hardest problems in **NP**
- If any of these hard problems are shown to be in **P**, then **P** = **NP**.
- Most "natural" problems are either in **P** or in the collection of the hardest problems in **NP**.



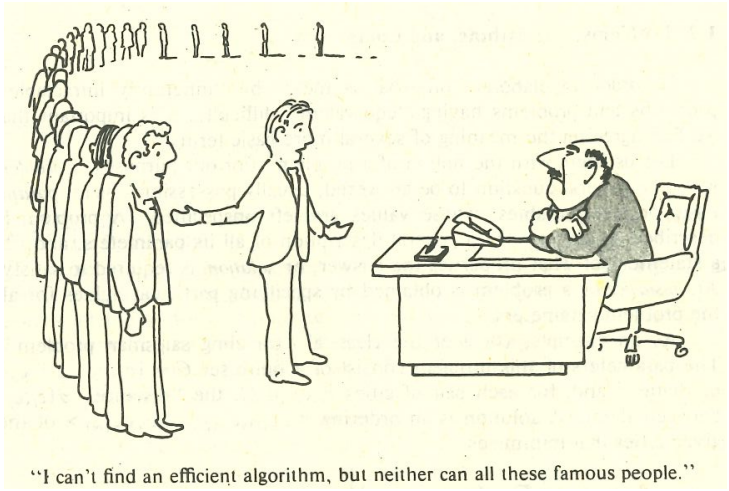
"I can't find an efficient algorithm, I guess I'm just too dumb."

Computers and Intractability by Garey & Johnson



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

Computers and Intractability by Garey & Johnson



Computers and Intractability by Garey & Johnson

Uncomputable Problems

There are problems which cannot be solved by any computer that will ever be built!

Examples:

- Given source code determine if the program halts
- Write the longest running program using *n* lines