Announcements

- Final exam: Friday, May 20, 9:30 AM, Chemistry 123
- Sample finals available online
- Review sessions: a) Wednesday 5/18 at 2:00PM in TBL 202
  b) Thursday 5/19 at 10 in TPL 205.
How Hard is Hard?
NP = Nondeterministic Polynomial

Partition
Subset Sum
Dinner Menu (3D-Matching)
Crossword
Longest Path
Traveling Salesman
Hamiltonian Path
Knapsack Problem
Minimum Graph Coloring

...
BENEDICT CUMBERBATCH IS OUTSTANDING

THE BEST BRITISH FILM OF THE YEAR

AN INSTANT CLASSIC

A SUPERB THRILLER

THE IMITATION GAME

BENEDICT CUMBERBATCH  KEIRA KNIGHTLEY

BASED ON THE INCREDIBLE TRUE STORY

IN CINEMAS NOVEMBER 14
Alan Mathison Turing
1912-1954
Father of Computer Science
Mathematician, Logician
Wartime Codebreaker
Victim of Prejudice

“Mathematics, rightly viewed, possesses not only truth but supreme beauty, a beauty cold and austere like that of sculpture.” - Bertrand Russell
The Halting Problem
(i.e., the not looping problem)

Write a program in some programming language that can analyze other programs in the same language and correctly decide whether or not the analyzed program will eventually stop running.
Goldbach's conjecture (1742)

Every even integer greater than 2 can be written as the sum of two primes.
Goldbach's conjecture

```java
int candidate = 4;

boolean counterExampleFound = false;
while ( ! counterExampleFound ) {
    candidate = candidate + 2;

    counterExampleFound = true;
    for ( int p = 2; p < candidate/2+1 && counterExampleFound ; p++ ) {
        if ( prime( p ) && prime( candidate - p ) ) {
            counterExampleFound = false;
        }
    }
}
```
The Halting Problem
(i.e., the not looping problem)

It is **impossible** to write a program in any reasonable programming language that can analyze another program in the same language and accurately decide whether or not the analyzed program eventually stops running.
ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine. In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes
Atanasoff-Berry Computer - 1942
Final Words

- Understanding the Internet
- Learning to program
  - a useful skill
  - understanding capabilities of computer
  - algorithmic thinking
- Glimpses of Computer Science
  - techniques for representing information
  - controlling complexity (layering)
  - discovery of algorithms
  - analysis/evaluation of algorithms
  - fundamental limits of computation