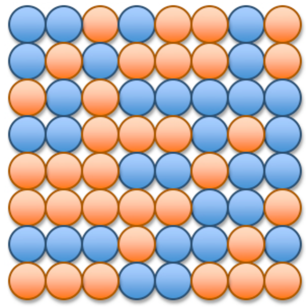


UNIT 14B
The Limits of Computing: P and NP

Decision Problems

- We have seen four examples of decision problems with simple brute-force algorithms that are intractable.

  – The Monkey Puzzle O(N!)
  – Traveling Salesperson O(N!)

  – Map Coloring $O(3^N)$
  – Satisfiability $O(2^N)$

Are These Problems Tractable?

- For any one of these problems, is there a single tractable (polynomial) algorithm to solve any instance of the problem?
  - Computer scientists have not been able to prove that general tractable algorithms exist for these problems and we just haven't found them yet.
  - Computer scientists have not been able to prove that general tractable algorithms do not exist for these problems so we should stop looking for these algorithms.

P and NP

- The class **P** consists of all those decision problems that can be solved on a deterministic sequential machine in an amount of time that is polynomial in the size of the input
- The class **NP** consists of all those decision problems whose positive solutions can be verified in polynomial time given the right information, or equivalently, whose solution can be found in polynomial time on a non-deterministic machine.

from Wikipedia

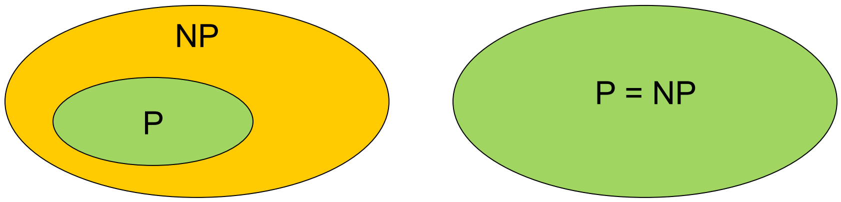
Example

- Finding the Minimum in an Array
  Solvable in polynomial time yes
  Verifiable in polynomial time yes
- Map Coloring
  Verifiable in polynomial time yes
  Solvable in polynomial time ?
- If a problem is in P, it must also be in NP.
- If a problem is in NP, is it also in P?

NP Complete

- The class **NPC** consists of all those problems in NP that are least likely to be in P.
  - Each of these problems is called **NP Complete**.
  - Monkey puzzle, Traveling salesperson, map coloring, and satisfiability are all in NPC.
- Every problem in NPC can be transformed to another problem in NPC.
  - If there were some way to solve one of these problems in polynomial time, we should be able to solve all of these problems in polynomial time.

Complexity Classes



If P ≠ NP, then some decision problems can't be solved in polynomial time.
If P = NP, then all computable problems can be solved in polynomial time.

The Clay Mathematics Institute is offering a $1M prize for the first person to prove P = NP or P ≠ NP.
(http://www.claymath.org/millennium/P_vs_NP/)

Watch out, Homer!

What’s Next?

- Are all computational problems solvable by computer?
  - NO!
    There are some that we can’t solve no matter how much time we give the computer, no matter how powerful the computer is.