EECS3101 notes:: Intractability

Jerry Wu

2023-04-04

The million dollar question

Abstract

Say we have a set P which contains all decision based problems that can be solved in a polynomial time on a deterministic Turing machine. For example, if the shortest path between two vertices on a graph is shorter than some value k. This implies that we are able to write a program to solve the problem.

Then we have a set NP which contains all decision based problems that are solvable in polynomial time in a non deterministic Turing machine. The whole point of this is to know when a problem **cannot** be solved efficiently (anything $\geq O(2^n)$). But is $P = NP$? If we can find a polynomial time solution for an NP complete problem, then $P = NP$. However, if we can find a problem that is in NP but not P, then $P \neq NP$.

"I have been wondering this for 50 years already"-Larry YL Zhang 2023

NP completeness

There are two classes of NP complete problems::

- **NP-hard**
  - If it can be solved efficiently, then it can be used as a subroutine to solve any other problem in NP efficiently.
- **NP-complete**
  - Problems that are both NP and NP-hard
  - This class is the hardest class of computational problems in NP.

"There are people who claim to have solved P=NP, but they are all pseudo-scientists"-Larry YL Zhang 2023

But what's the point?

The reason why we want to find NP completeness is so that we can save time from finding a polynomial solution. We do this throigh proving that a polynomial time solution doesn't exist.

> *"Keep trying to prove it, and you might win a million dollats"-Larry YL Zhang 2023*

NP complete examples

Satisfiability problem (SAT)

Given a boolean formula written in a canonical conjunction of disjunctions form, decide if there exists a set of values of $x_1, x_2, \ldots, x_n$ that makes the expression evaluate to TRUE. A 2SAT problem (2 variables) is a P complete problem, and the 3SAT problem (3 variables) is proven to be NP complete.

The travelling salesman problem (TSP)

- A travelling salesperson needs to visit n cities.
- Is there a route of at most d length? (decision)
- Optomization problem:: find a shortest cycle visiting all vertices once in a weighted graph

TSP is proven to be NP complete, as well as the longest path in a graph.

CLIQUE

A clique is a subgraph of a graph where all the vertices are adjacent to each other.

- Given an undirected graph with n vertices, is there a subset of size s such that all pairs of vertices in the subset are adjacent to each other?

Identifying NP completeness

We can do this by way of reduction.

- If problem B can be solved by solving an instance of A, i.e. A is harder than B, then we can say B reduces to A. This is counter intuitive since A is harder than B.
- If we can find an algorithm that reduces B to A in polynomial time, and that B is NP complete, then A is also NP complete.

Example

- SAT actually reduces to CLIQUE
- Given any input of SAT, we can convert it to an input of a CLIQUE problem.
- For an SAT formula with k disjunctions, we construct a CLIQUE input that has a clique of size K iff the original boolean formula is satisfiable.

We can represent each vertex of the graph as each literal that appears in any given boolean formula. Each edge will represent two vertices that are connected unless::

- They come from the same disjunction
- They represent X and $\neg X$ for some X
- What happens if we find a clique of size 4? How does it relate to the SAT solution?

Algorithm approximations

Approximate when finding a polynomial solution isn't realistic. It is possible to prove that the approximate solution tends towards the optimal solution.

Summary

Today we only scratched the surface of this topic. Take 4111 or 4115 if you want to know more! (not me)