H A R D

It's too hard.

Daiwei Chen

April 19, 2019

Contents

1 Hard Problems

Many problems have polynomial $O(n^k)$ time solutions. Some problems haven't had a polynomial time solution.

2 Complexity Classes

P Set of problems we can solve in polynomial time. Sorting, Searching...

NP Verifiable in polynomial time. If you can solve in polynomial time, you can verify in polynomial time, but it's not always true in the other direction.

NP-Complete Set of problems that ANY problem in NP can be transformed into in polynomial time. Decision.

NP-Hard At least as hard as hard as hard as hard as any problem in NP. Optimization.

3 P vs NP

P = NP? If you can find a polynomial solution in NP-Complete, then you can transform any NP problem into NP-Complete in polynomial time and then solve it. So if you can prove P = NP you're a genius millionare.

4 NP-Complete Problems

4.1 Boolean Satisfiability

kCNF (k Conjunctive Normal Form) - A way to express a logical statement. Each or has exactly k variables.

- "AND" of many "OR" expressions, each of which contains K variables or their negations.
- (P || !Q) && (R || Q) && (!P || !R) - 2 CNF because each or has 2 variables.

4.2 kSAT

Given an expression in KCNF is there an assignment of vars that makes it evaluate to true.
k = 2 (2SAT) ϵ P
k > 2 ϵ NP-Complete

5 Graph Coloring

- Assign colors to nodes such that no two adjacent nodes have the same color
- Is it possible to color a graph with t colors
- if t=2 P, else if t > 2 NP-Complete
- Finding the minimum chromatic number is NP-Hard

6 Hamiltonian Paths / Cycles

- Given a graph is there a simple path (or cycle) that visits every vertex? Visit every single vertex once AND get back to the starting point.
- NP-Complete

6.1 k-SAT -> Hamiltonian Path

E1 (P || !Q) &&

E2 (Q || R) &&

E3 (!P || !R)

P, Q, R, craft the *Hamburger of doom*

When you are checking, you must check if you can visit every node without repeating a path.

2-SAT is P

Above is NP-Complete

7 Euler Tour / Path

- Use every edge once
- Look at every verticie with an odd degree, if each one has an odd number of edges attached to it, then you will have an Euler Path. Just check for even or odd number of edges attached to the node.
- P

8 Traveling Salesman Problem

- NP-Hard

9 Bin Packing

- NP-Hard
- Given a list of items & bins of a standard size, what is the minimal number of bins needed to store the items?
- For example, if you have a bunch of songs to burn and a fixed capacity for CDs how will you fit it in?

C = 10, S = [5, 6, 3, 7, 5, 4]
Best fit: 3 Bins Opt(L): Optimal # of bins

9.1 Online

Items handled in order that they arrived in. The benefit is speed, so if it's a truck packing problem then the trucks can be sent out and not wait there.

9.1.1 Next Fit

- Next(L) $\leq$ 2 * Opt(L) - 1

If the next item fits in the current bin, put it in there, if not, close the bin and open a new one.

Next Fit	
B1	5
B2	6, 3
B3	7
B4	5, 4

9.1.2 Frist Fit

- First(l) $\leq$ ceil(17/10 * Opt(L))

Keep bins open until full, put item in 1st bin it fits in

First Fit	
B1	5, 3
B2	6, 4
B3	7
B4	5

9.2 Offline

All items obtained and then fitted.

9.2.1 First Fit Decreasing

- FFD(l) $\leq$ ceil(11/9 * Opt(L))

Sort Items by decreasing size, then do first fit.

First Fit Decreasing	
B1	7, 3
B2	6, 4
B3	5, 5