7 March 2024

P: problems solvable in polynomial time

$O(n^k)$ time, n is the size of the problem
k is fixed constant

NP: problems whose solutions are verifiable in polynomial time

SATISFIABILITY PROBLEM:

Let $a, b, c, \ldots$ be boolean variables (take on True/False values).

Given a statement consisting of these variables and operators OR, AND, NOT, is there an assignment of True/False values that satisfies the statement?

Example: (a OR b OR NOT c) AND (NOT b OR C)

a,b,c all True: $\underbrace{(T \text{ or } T \text{ or } F)}_{T}$ and $\underbrace{(F \text{ or } T)}_{}$

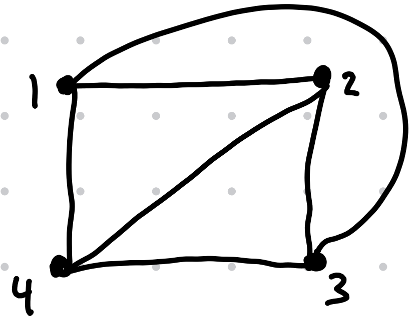
T and T = T

Special Case: 3-SAT

each OR clause has exactly 3 variables

Graph coloring problem <u>reduces</u> to satisfiability problem. (If you can solve satisfiability, then you can solve graph coloring.)

<u>Example:</u>

G: 

Is G 3-colorable?

Define satisfiability clauses involving 12 Boolean variables. $x_{i,j}$ where $x_{i,j}$ = True iff vertex i has color j. $i \in \{1,2,3,4\}$, $j \in \{1,2,3\}$

Clauses: $C_i = \underbrace{(x_{i,1} \text{ OR } x_{i,2} \text{ OR } x_{i,3})}_{\text{vertex } i \text{ has at least one color}}$ for $i \in \{1,2,3,4\}$

NOT$(x_{i,1}$ AND $x_{i,2})$ = $T_i = (\text{NOT } x_{i,1} \text{ OR } \text{NOT } x_{i,2})$ — can't have both colors 1 and 2

$U_i = (\text{NOT } x_{i,1} \text{ OR } \text{NOT } x_{i,3})$

$V_i = (\text{NOT } x_{i,2} \text{ OR } \text{NOT } x_{i,3})$

vertex i has at most 1 color

For each edge e of G, we must ensure that the endpoints of e are not the same color

Suppose endpoints of e are u and v:

let $D_{e,j} = (\text{NOT } x_{u,j} \text{ OR } \text{NOT } x_{v,j})$

$j \in \{1,2,3\}$

(6 edges)(3 colors) = 18 clauses

Graph G is 3-colorable iff <u>all</u> of these clauses can be satisfied simultaneously.

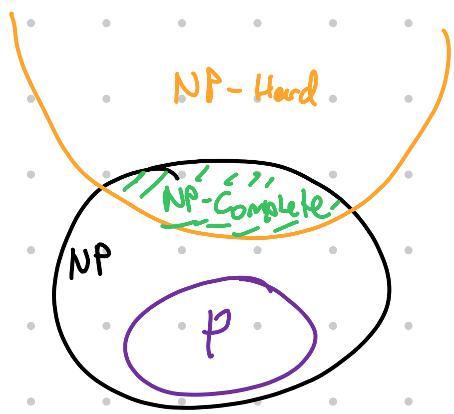
NP-Hard: Problem H is NP-hard if every other problem in NP can be reduced to H in polynomial time.

→ If you can solve H in polynomial time, then you can solve all problems in NP in polynomial time.

NP-Complete: Problem H is NP-complete if:

(1) H is in NP

(2) Every problem in NP is reducible to H in polynomial time.



Traveling Salesperson Problem (TSP)

Given a collection of cities and distances between them, what is the shortest path that visits all of the cities and returns to the starting point?

$n = 10$ cities: say we start at 1

then any permutation of $\{2, 3, 4, \dots, 10\}$ gives a tour

ordering

There are $9!$ such permutations

Heap's Algorithm lists all permutations of n items