CS 673 Algorithms

KT 8.1~8.4

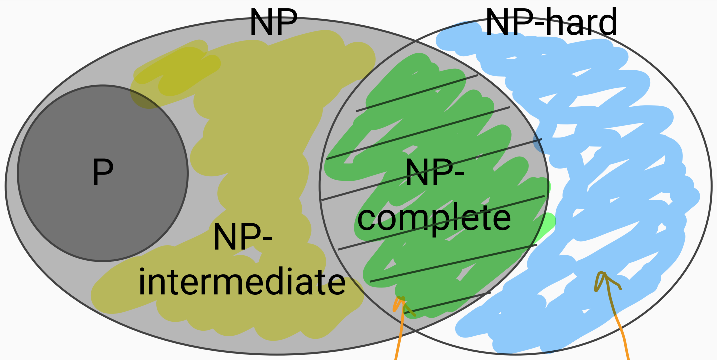
Recap: Chapter 8 NP & Intractability

- KT Chapter 8.1, 8.2, & 8.3
  - Poly-time reductions (formal definition) & Definition of class P & NP
  - 3-SAT $\leq_P$ Independent Set $\leq_P$ Vertex Cover
- KT Chapter 8.4
  - NP-hardness and NP-completeness
  - Circuit Satisfiability: The first NP-complete problem
  - Circuit SAT $\leq_P$ 3-SAT $\leq_P$ Independent Set and so on
- KT Chapter 8.5-8.8 & 8.10
  - Most useful & well-known NP-Complete problems

X is NP-intermediate, Y ∈ NPC
$\Leftrightarrow$ $x \le_p Y$ but $\cancel{Y \le_p X}$
Y is atleast as difficult as 'X'!

'x' is NP-complete $\Leftrightarrow$ $x \in NP$, x is NP-hard

Classifying problems

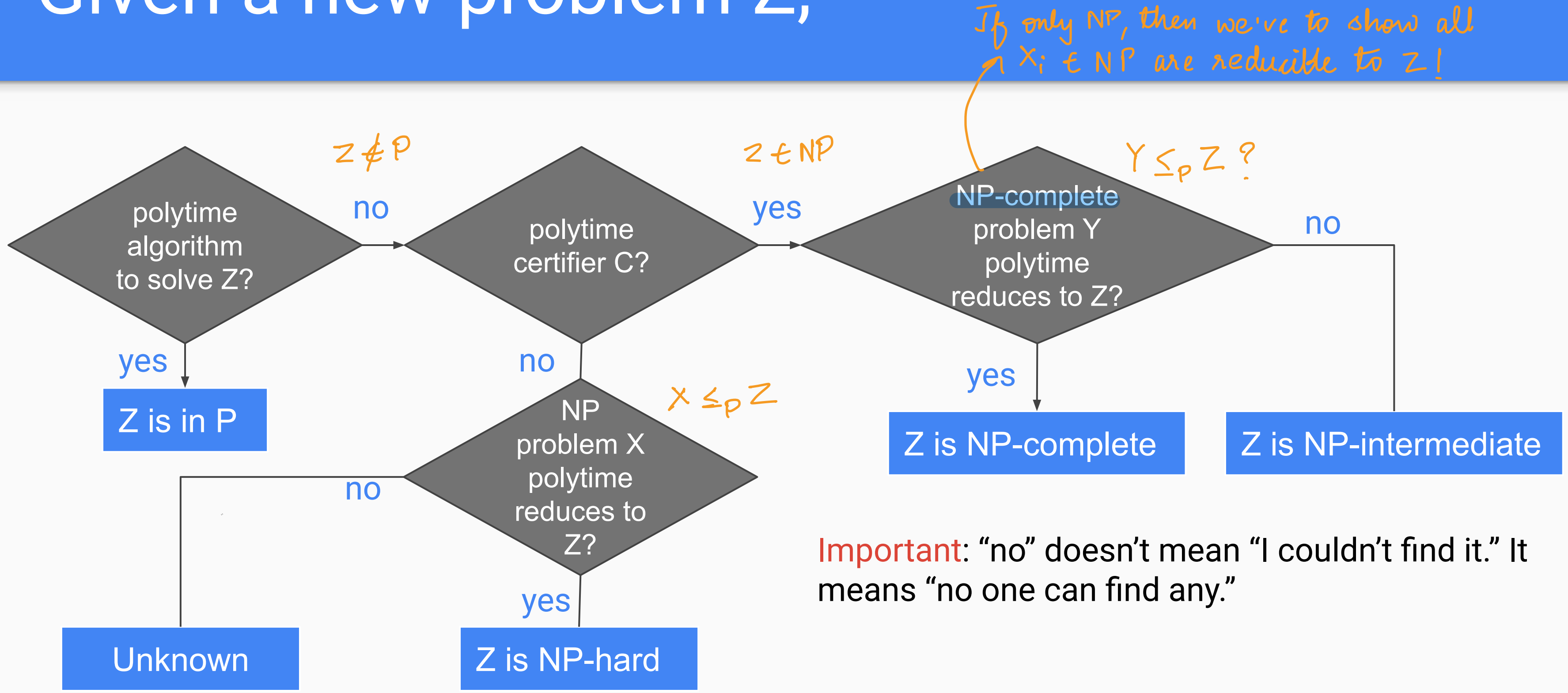


- Examples
  - Vertex Cover is NP-complete
  - Graph isomorphism is NP, but not NPC
  - Halting problem is NP-hard, but not NP

We only consider decision problems. For problem X,

- X is in P ⇔ known poly-time algorithm to solve X
- X is in NP ⇔ poly-time certifier C for (s, t), where s is an instance of X and t is a potential solution
- X is NP complete ⇔ any problem Y in NP poly-time reduces to X (Y <=p X, X, Y are in NP)
- X is NP hard ⇔ X is as hard as any problem Y in NP (Y <=p X, Y in NP, X may or may not be in NP)
- X is NP-intermediate ⇔ X is in NP but not P nor NPC

Given a new problem Z,



Important: “no” doesn’t mean “I couldn’t find it.” It means “no one can find any.”

Comparing problems' difficulties

X≤pY ⇔ X poly-time reduces to Y ⇔ Y is as hard as X

In the ascending order of difficulties: P, NP-intermediate, NP-complete, NP-hard

- If X is in P, Y can be
- If Y is NP-hard, X can be
- If X is in NP, Y can be
- If Y is in NP, X can be

NP-Completeness

- To prove that a new problem (Y) is NP-complete, you must do three things:
  - Prove that Y belongs to NP (easy)
  - Pick a known NP-complete problem X (tricky) – find Problem with similar outcome!
  - Show that X $\leq_P$ Y (relatively easy if the right problem was picked in step 2)

- How to pick the right problem
  - KT Chapter 8.10 categorizes NP-C problems based on their objectives & constraints.
  - Examine the objectives/constraints of Y, and try to pick the most similar problem as X.
  - (If nothing works, 3-SAT is often regarded as go-to problem since it is the most natural problem to reduce from.)

Recap: Chapter 7 Network Flow

- KT Chapter 7.1 & 7.2
  - Min Cut $\leq_P$ Max Flow
- KT Chapter 7.5
  - Bipartite Matching $\leq_P$ Max Flow
- KT Chapter 7.10
  - Image Segmentation $\leq_P$ Min Cut $\leq_P$ Max Flow

Recap: Chapter 8 NP & Intractability

- KT Chapter 8.1, 8.2, & 8.3
  - Poly-time reductions (formal definition) & Definition of class P & NP
  - 3-SAT $\leq_P$ Independent Set $\leq_P$ Vertex Cover
- KT Chapter 8.4
  - NP-hardness and NP-completeness
  - Circuit Satisfiability: The first NP-complete problem
  - Circuit SAT $\leq_P$ 3-SAT $\leq_P$ Independent Set and so on
- KT Chapter 8.5-8.8 & 8.10
  - Most useful & well-known NP-Complete problems

More examples

- Interval Scheduling $\leq_P$ Independent Set
- Weighted Interval Scheduling $\leq_P$ Weighted Independent Set

Readings & Exercises

- KT Chapter 8.* (except for Chapter 8.9)
  - You won't be asked to produce complex reductions, but you should read about the reductions provided in the textbook.
- KT Exercises

Easy: 8.2, 8.3, and 8.4

Medium: 8.5, 8.16, 8.22, and 8.28 (try at least two)

Optional: 8.17, 8.29, and 8.41

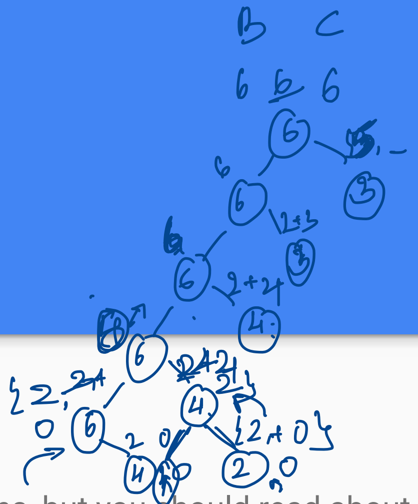
(optional) Karp's 21 NP-Complete Problems: https://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems

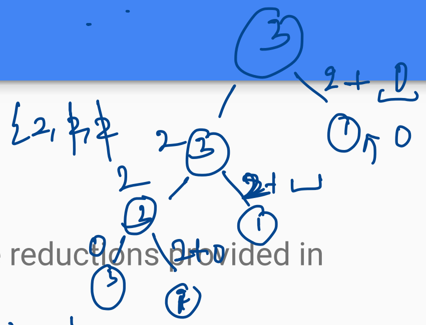
$w = \{2,2,2,3,3\}$

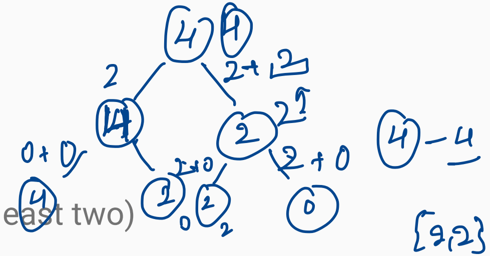
$n_2 = 3$

$n_3 = 2$

$\frac{12}{2} = 6$







$w = \{w_1, \ldots, w_n\}$

$v = \{v_1, \ldots, v_n\}$

$opt \Rightarrow V_{opt}$ Greedy $\Rightarrow V_G$

if for all instances of knapsack pblm

$\boxed{2 \cdot V_G \geq V_{opt} \geq V_G} \Rightarrow \boxed{\frac{V_{opt}}{V_G} \leq 2}$