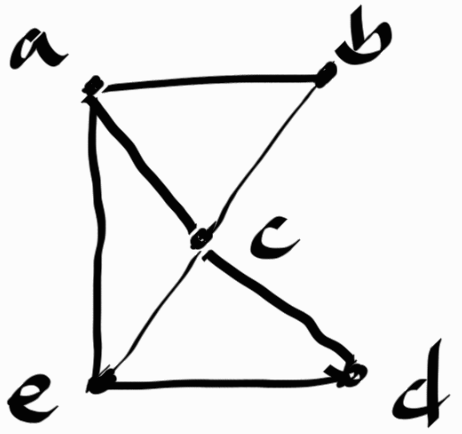
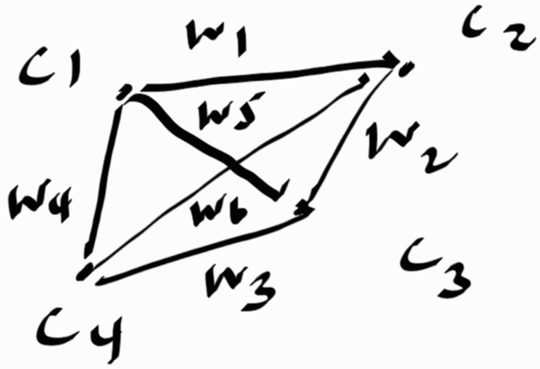
Example:

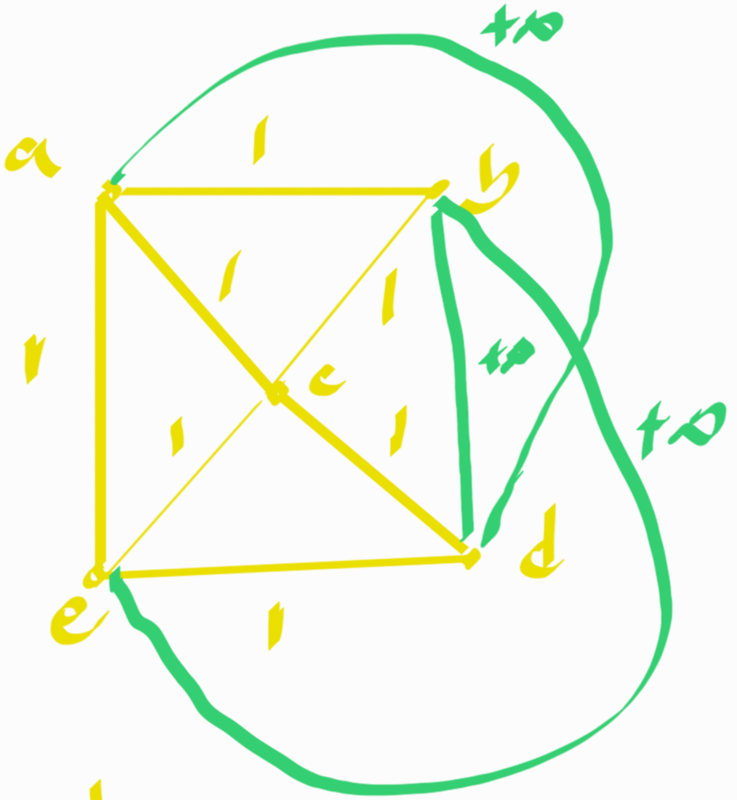
HC $\leq_p$ TSP (on complete weighted graphs)



$\langle a, b, c, d, e, a \rangle$ HC

G





$\langle a, b, c, d, e, a \rangle$

G'

Given an 'input' graph $G = (V, E)$, we construct graph $G' = (V', E', \omega)$, which is a complete weighted graph as follows.

- we keep the original vertices.
- we add new edges between vertices which we did not have edges.
- we give weights to edges as follows.

  $\forall e \in E'$

  If $e \in E$, $w(e) = 1$

  If $e \notin E$, $w(e) = +\infty$
- This transformation takes $O(n^2)$

- Claim! There is a HC in G iff there is a TSP with weight n in G'

only if $\Rightarrow$

If G has a HC, then use the edges in HC to form a TSP in G'.
As each edge in TSP has weight 1.
The total weight is n.

If part $\Leftarrow$

If G' has a TSP of size n,
we know that all the edges with weight two cannot be part of it.
Therefore, all edges in TSP must have weight 1, which in turn means are in G.
Then HC gives us a HC in G.

SET-PARTITION $\leq_p$ RECTANGLE PACKING

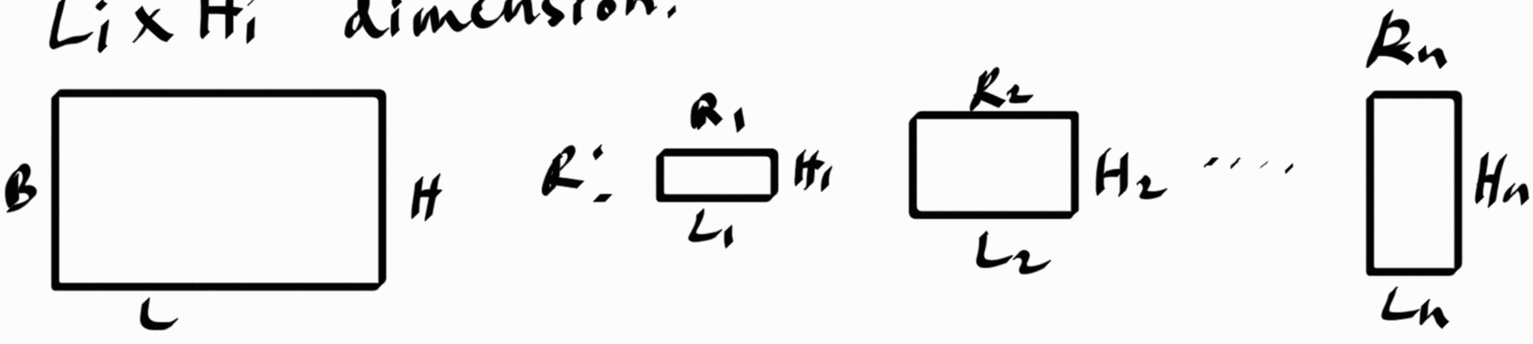
set partition problem is given a set of integers $S = \{a_1, a_2, \dots, a_n\}$, find whether there exists a subset $S' \subseteq S$ s.t $\sum_{x \in S'} x = \sum_{x \in S \setminus S'} x$

$S = \{8, 1, 1, 9, 11, 2, 3, 5\}$

$\underbrace{\{8, 1, 11\}}_{20}$ $\underbrace{\{1, 9, 2, 3, 5\}}_{20}$

Rectangle packing

Given a container B of size $L \times H$
You are given set of rectangles R
$R = \{R_1, R_2, \dots, R_n\}$, where R_i is of $L_i \times H_i$ dimension.



Reduction:

Let the input to the set partition be $S=\{a_1, a_2, a_3, \dots, a_n\}$

$\forall a_i$, construct R_i with length a_i and height ε

- B is constructed to have length $\frac{\sum a_i}{2}$ and hieght 2ε

* There is a solution for set partition instance iff all the R_i's can be packed into B

$S = \{8, 1, 1, 9, 11, 2, 3, 5\}$

reduction takes $O(n)$

Recap

What is NP-complete?

A problem B is NP-complete

1. $B \in NP$
2. Every problem in NP can be polynomially reducible to B.

Only the first NP-complete problem (SAT) is proved using the above definition.

---

we say B is NP-complete

1. $B \in NP$
2. $C \leq_p B$, where C is NP-complete

---

suppose A is any NP problem

$A \leq_p C \leq_p B$

What is NP-hard?

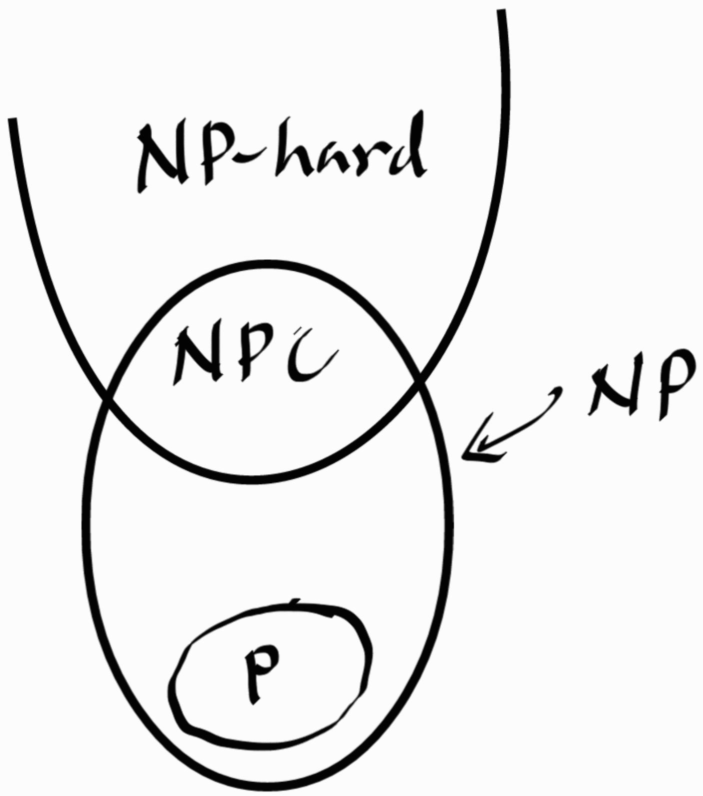
Suppose C is NPC.

Also suppose $C \leq_p A$.

If we donot show that A is in NP, then we cannot say A is NPC

But we can say A is NP-hard.

Usnaley optimization problems are NP-hard.



SAT (satisfiability problem) is NP-complete.

$$\text{Input } \phi = (x_1 \vee \boxed{x_2} \vee \bar{x}_3 \vee \boxed{x_4}) \wedge \overbrace{(x_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_1)}^{\text{clauses}}$$

$$\wedge \underbrace{(x_7 \vee \bar{x}_1)}_{\text{clauses}} \ldots\ldots$$

Q: Is there a truth assignment that satisfies the ϕ

NP-complete $O(2^n)$

$\boxed{\text{3-SAT}}$: A special case of SAT problem, where every clause has 3 literals.

Ex: $\phi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_5 \vee x_6) \wedge \ldots$

$\ldots \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_5)$

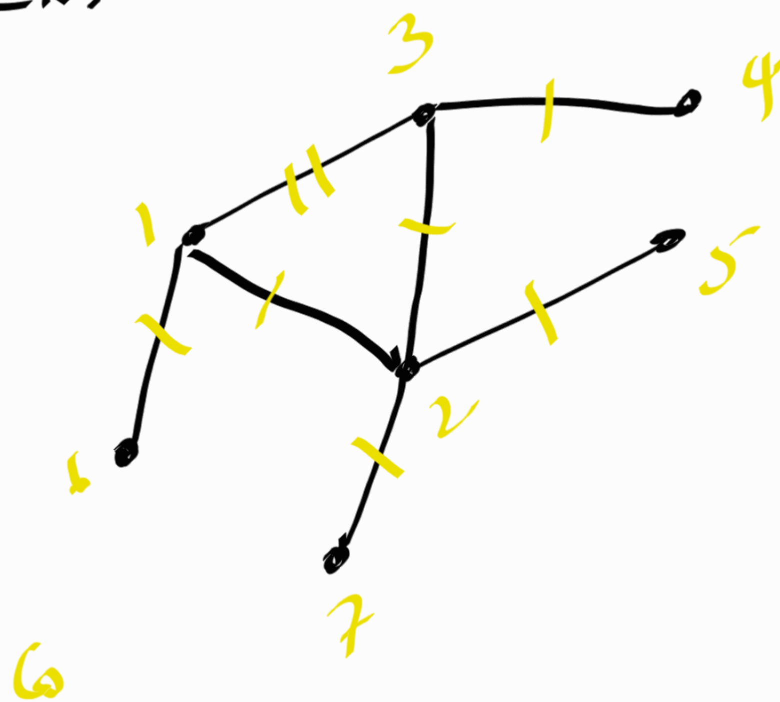
Let's try to learn how to show a problem is NP-complete.

We want to show vertex-cover problem is NP-complete.

What is vertex-cover?

Given a graph $G = (V, E)$, find the minimum number of vertices that covers all edges.

Ex:



$\{1, 3, 2$

This is an optimization problem. Let's convert this to a decision problem.

Given Graph $G=(V,E)$ and $k \in \mathbb{N}$, is there a vertex cover of G at most k?

Think about the definition of NP-completeness.

1. The problem is in NP.
2. Every problem in NP is polynomially reducible to the problem.

Proof: Let's prove VC $\in$ NP.

Sketch:

1. VC $\in$ NP
2. 3SAT $\leq_p$ VC

Let's create an NTM that decides VC.

N = on input $\langle G, k \rangle$

1. Nondeterministically select k nodes from G.
2. Delete the k nodes and edges incident to it. If there are edges left, reject; otherwise accept.

Clearly deletion of the nodes and edges can be done in $O(n^2)$.

Second we need to show any problem is NP can be polynomially reducible to VC.

$$3SAT \leq_p VC$$