HW11 due Tue Dec 2nd.

GPS 11 due Mon Dec 8th

No HW Party either Sunday of break.

Today + Mon 1st still on.

FLEX evaluations Sat Nov 29 to Thur Dec 11th

TAC ok on Sat Dec 13.

Survey about 374 0.5% EC if you fill it.

see website, Discord, etc. over weekend.

Subset Sum:

Given a set X of positive integers &
one more integer T.
nonnegation

Is there a subset of X summing to T?

$\{2, 5, 10\}$ $\quad T = 15$ ✓

$\qquad\qquad\qquad T = 9$ ✗

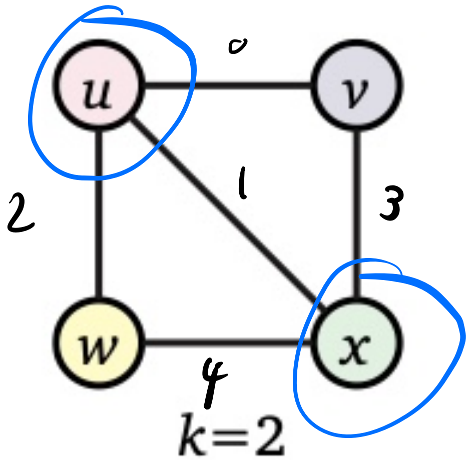
SubsetSum $\in$ NP.

SubsetSum is NP-hard.

Reduce from VertexCover.

Given an undirected $G=(V,E)$ + an integer k.
Is there a vertex cover of size k.

Build a SubsetSum instance using gadgets



Number the edge from 0 to $|E|-1$

For each edge i, add edge gadget

$$b_i := 4^i \to X$$

For each vertex v, let $\Delta(v)$ be its incident edges.

$$a_v := 4^{|E|} + \sum_{i \in \Delta(v)} 4^i \to X$$

$b_0 = 000001_4$

$b_1 = 000010_4$

$b_2 = 000100_4$

$b_3 = 001000_4$

$b_4 = 010000_4$

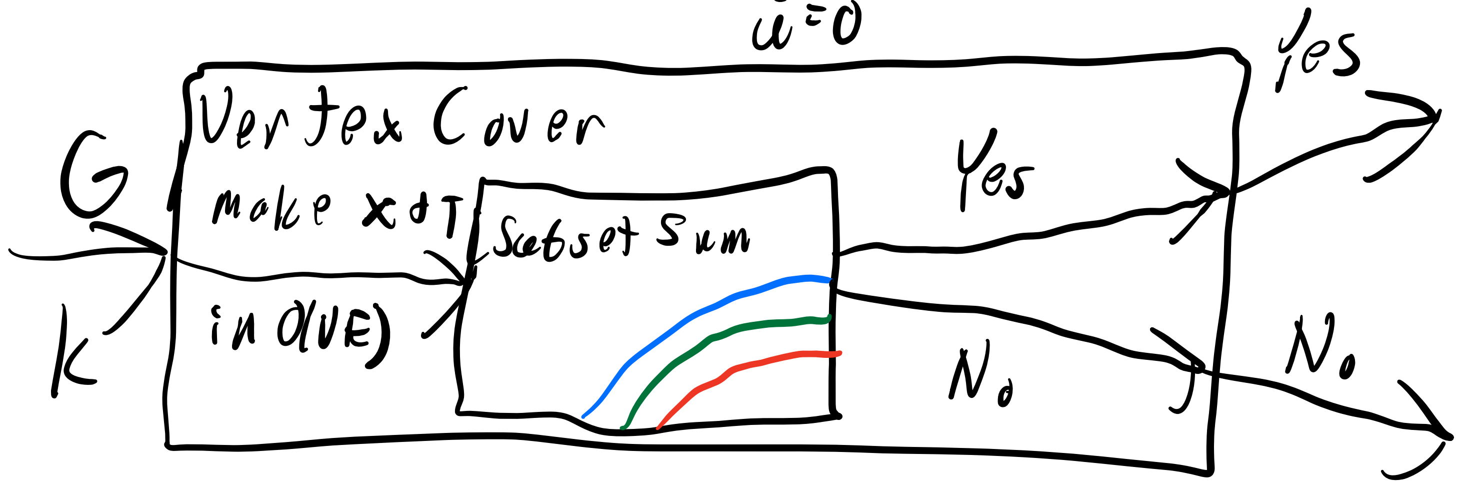
$a_u = 100111_4 \qquad a_v = 101001_4$

$a_w = 110100_4 \qquad a_x = 111010_4$

$$T := k \cdot 4^{|E|} + \sum_{i=0}^{|E|-1} 2 \cdot 4^i$$

$T = k22222_4$

$k \cdot 4^{|E|}$



G has a vertex cover of size k iff there is a subset of X summing to T

$\Rightarrow$ Suppose G has a vertex cover C of size k.

Let $X' = \{a_v \mid v \in C\} \cup \{b_i \mid$ edge i has $\underline{\underline{\text{one}}}$ endpoint in $C\}$.

For $i = |E|$, crumb $i = 2$. (no carries from ≤ 3 1's in base 4)

k vertex gadget add in another $k \cdot 4^{|E|}$

So $\sum_{x \in X'} x = T$.

$\Leftarrow$ Suppose some $X' \subseteq X$ sums to T.

There is $V' \subseteq V$, $E' \subseteq E$ s.t.

$$\sum_{a_v \in V'} a_v + \sum_{i \in E'} b_i = T.$$

No carries early, so we had to use at least one v incident to each edge i.

& we took k a_v's to get $k \cdot 4^{|E|}$

so V' is a vertex cover.

Subset Sum has an $O(nT)$ time algorithm

T: exponential in input size (number of <u>bits</u>) ?!

<u>Pseudo-polynomial time</u>.

<u>Weakly NP-hard</u>: Reductions created numbers exponential in input size

<u>Strongly NP-hard</u>: Any numbers can be written in unary during reductions.

Why? More practice transforming problems.

Know when to "give up".

- is there another approach
- can you specialize?
- approximation algorithms
- heuristics that work in practice?

NP-hard problems,

3SAT ← binary choices

Max Ind Set ← large subsets

Max Clique

Min Vertex Cover ← small subsets

3Color ← Partition objects into

Min Color   3 or more groups

Directed/Undirected Ham Cycle/Path ← order

or subgraphs

Subset Sum   "balanced subsets"

Partition - partition a set of integers into

two subset with equal sum

3DMatching/Exact3DMatching/X3M -
Given three subsets A, B, & C of n items each & a subset of $A \times B \times C$.
Pick exactly n triples from the subset to cover A, B, & C.

3Partion - Partion a set of integers into subsets of size 3 each summing to same value.

To prove Y is NP-hard, reduce arbitrary inputs x of a known NP-hard problem X to some input y of Y.

i) pick the most constrained X you

can

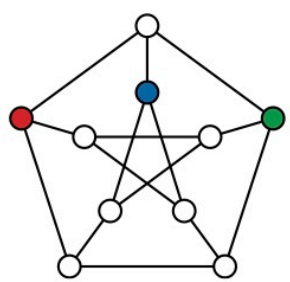
2) make as specialized an instance of Y as you can.

(d) Given a graph G with weighted edges and an integer ℓ, compute the minimum-weight spanning tree with at most ℓ leaves.

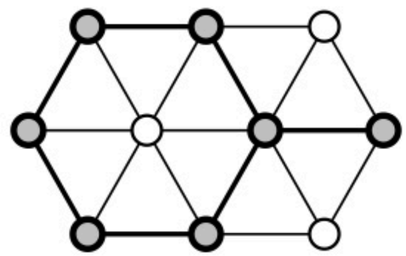
Ham Path $\ell = 2$

26. Let $G = (V, E)$ be a graph. A *dominating set* in G is a subset S of the vertices such that every vertex in G is either in S or adjacent to a vertex in S. The DOMINATINGSET problem asks, given a graph G and an integer k as input, whether G contains a dominating set of size k. Prove that this problem is NP-hard.

Min Vertex Cover



27. A subset S of vertices in an undirected graph G is *triangle-free* if, for every triple of vertices $u, v, w \in S$, at least one of the three edges uv, uw, vw is *absent* from G. Prove that finding the size of the largest triangle-free subset of vertices in a given undirected graph is NP-hard.



Max Ind Set

36. Jeff tries to make his students happy. At the beginning of class, he passes out a questionnaire that lists a number of possible course policies in areas where

he is flexible. Every student is asked to respond to each possible course policy with one of "strongly favor", "mostly neutral", or "strongly oppose". Each student may respond with "strongly favor" or "strongly oppose" to at most five questions. Because Jeff's students are very understanding, each student is happy if (but only if) he or she prevails in at least one of their strong policy preferences. Either describe a polynomial-time algorithm for setting course policy to maximize the number of happy students, or show that the problem is NP-hard.

3SAT

Yes or no on many policies