SUBSET SUM (S.S) Decision Problem

Given a finite set of positive integers

$S = \{w1, w2, \ldots, w_n\}$ & int t, is there a subset

$S' \subseteq S$ whose sum equals t?

i.e. $S' \subseteq S$ such that $\sum_{w \in S'} w = t$?

<u>example</u> $S = \{3, 4, 5, 11, 13, 21, 34\}$, $t = 33$?

Yes. $S' = \{4, 5, 11, 13\}$

Will prove SS is in NP-Complete.

For the reduction portion of the proof, we'll show Vertex Cover $\leq_p$ S.S.

Theorem: Subset Sum (Decision) Problem is NP-Complete

PROOF:

(1) S.S. is in NP.

Verification PROBLEM

Given set of ints S, int t and a set S', verify that S' is a subset of S & the elements of S' sum to t.

- Give an algorithm to verify that S' is a subset of S & explain why this algo runs in polynomial time. Will leave to the reader -- or talk about it in class. And also verify the sum of the ints in S' will be t.

SS. is in NP-Complete

Proof continued.

2. SS is in NP-Hard

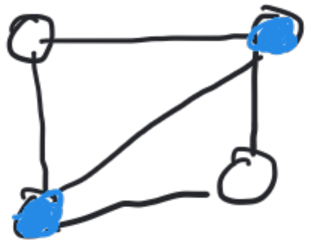
For the reduction portion, we'll Vertex Cover $\leq_P$ S.S.

Claim: Assume by contradiction that there is a polynomial time algorithm A for solving SS, that, given a finite set of positive integers S and a target integer t> 0, returns whether there is a subset S' of S whose elements sum to t.

Vertex Cover Decision Problem: Given a graph G & int $K > 0$, can edges be covered by K vertices?

Algo: VC (G, k)
Inp: Graph G & integer $K > 0$
Output: Return true if there is a vertex cover of size K, false ⓔ

// we'll use Algo A to Decide VC
Invoke Algo A for subset sum (Decision) problem passing in S & t



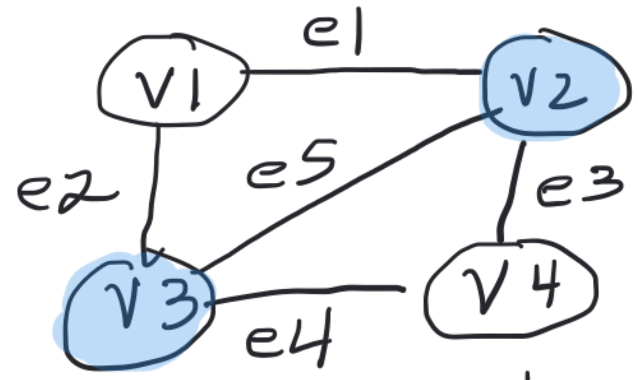
G

Recall: V.C. Definition
V.C. = set of vertices s.t. each edge in G is incident to at least 1 vertex in the set.

	What it does	What we want
V.C.	SELECTS VERTICES	cover all edges
SS	SELECTS INTEGERS	Determine a target sum t

Incidence Matrix H

	e_1	e_2	e_3	e_4	e_5
v_1	1	1	0	0	0
v_2	1	0	1	0	1
v_3	0	1	0	1	1
v_4	0	0	1	1	0



$|E| = m, \quad |V| = n$

$$w(v_i) = 4^{m+1} + \sum_{j=1}^{m} H[i,j] \cdot 4^j$$

This is the weight or value assigned to vertex v_i

$$w(e_j) = 4^j \text{ for each edge } e_j$$

edge $e5$ is covered twice

$w(v_1) = 4^6 + 4^1 + 4^2$

$w(v_2) = 4^6 + 4^1 + 4^3 + 4^5 = 8096 + 4 + 64 + 1024 = 9188$

$w(v_3) = 4^6 + 4^2 + 4^4 + 4^5 = 8096 + 16 + 256 + 1024 = 9392$

$w(v_4) = 4^6 + 4^3 + 4^4$

$w(e_1) = 4^1 = 4$

$w(e_2) = 4^2 = 16$

$w(e_3) = 4^3 = 64$

$w(e_4) = 4^4 = 256$

$w(e_5) = 4^5 = 1024$

$t = k \cdot 4^{m+1} + \sum_{j=1}^{m} 2 \cdot 4^{j}$, k = # of vertices in V.C.

but only use within this calculation the vertices that are in the vertex cover & consider the target sum.

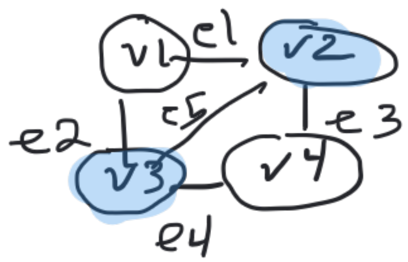
$t = 2 \cdot 4^{6} + 2 \cdot 4^{1} + 2 \cdot 4^{2} + 2 \cdot 4^{3} + 2 \cdot 4^{4} + 2 \cdot 4^{5}$

$= 2 \cdot 8096 + 2 \cdot 4 + 2 \cdot 16 + 2 \cdot 64 + 2 \cdot 256 + 2 \cdot 1024$

$= 16192 + 8 + 32 + 128 + 512 + 2048$

$= 18920$

To include in set S that algo A uses in the subset sum,



- include $w(e_j)$ in S if e_j is covered by only 1 vertex
- include $w(v_i)$ in S if vertex v_i is in V.C.

$S = \{w(v_2), w(v_3), w(e_1), w(e_2), w(e_3), w(e_4)\}$

$= \{9188, 9392, 4, 16, 64, 256\}$

If you sum these you get $18920 = t$

Notice, $w(e_5)$ is not included in S Because it is covered by 2 vertices.

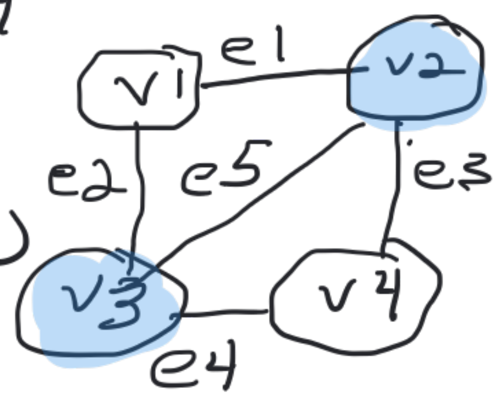
Why this works? $\boxed{w(e_j) = 4^j}$

Recall: $\boxed{w(v_i) = 4^{m+1} + \sum_{j=1}^{m} H[i,j]\cdot 4^j}$, $\boxed{t = k\cdot 4^{m+1} + \sum_{j=1}^{m} 2\cdot 4^j}$

In our example $t = \overset{\text{term 1}}{\boxed{2\cdot 4^6}} + \overset{\text{term 2}}{\boxed{2\cdot 4^1}} + \overset{\text{term 3}}{\boxed{2\cdot 4^2}} + \overset{\text{term 4}}{\boxed{2\cdot 4^3}} + \overset{\text{term 5}}{\boxed{2\cdot 4^4}} + \overset{\text{term 6}}{\boxed{2\cdot 4^5}}$

$S = \{w(v_2), w(v_3), w(e_1), w(e_2), w(e_3), w(e_4)\}$

$= \{9188, 9324, 4^1, 4^2, 4^3, 4^4\}$



- In a v.c. each edge is covered by 1 or 2 vertices
- In calculating t

$\boxed{\text{term 1}}$ we have $k = 2$ vertices in our v.c.

In calculating the value for each vertex $w(v_i)$ we add 4^{m+1} to its value.

In calculating the target sum t we include $4^{m+1}\cdot k$

In calculating the $\boxed{\text{Remaining term}}$ correspond to The edge weights.

$V.C. \leq_p S.S.$

proof of correctness:

($\rightarrow$) suppose a graph G has a V.C. $= \{v_1, v_2, \ldots, v_k\}$ of size k.

- In a V.C. each edge is covered by 1 or 2 vertices.
- In calculating our target sum t, $t = k \cdot 4^{m+1} + \sum_{j=1}^{m} 2 \cdot 4^j$. The leading term accounts for the # of vertices k in our V.C. multiplied by 4^{m+1}. The remaining terms in t's formula correspond to the m edges of the graph, where each value assigned to the edge is multiplied by 2, since the edge can be covered by at most 2 vertices.
- In calculating the set of integers S, we include the value assigned to each vertex in the V.C., $w(v_i) = 4^{m+1} + \sum_{j=1}^{m} H[i,j] \cdot 4^j$. $H[i,j]$ is 1 if vertex v_i is incident to edge e_j and zero otherwise. Only the values for vertices in the V.C. are included in S. Additionally, the value for the edges that are covered by a single vertex are placed is S, since an edge that is covered by 2 vertices is included within the value calculated for each of these 2 vertices. $\square$

($\Leftarrow$) proof of correctness continued.

Suppose there is a subset of integers that sum to t. Since t contains k values of 4^{m+1} within its calculation, it must include exactly k of the vertex values. So, we will include vertex v_i in our cover for each corresponding $w(v_i)$ value.

This set is a cover because each edge e_j corresponds to a power 4^j & must contribute 2 values in this sum. If only one of these values 4^j came from $w(e_j) = 4^j$, the other must have come from one of the chosen $w(v_i)$. $\square$