week11_lec1

Main Topics :

- NP-Completeness
- The Vertex Cover Problem
- Subset Sum

NP- Completeness

A language B is NP-complete if:

- B is in NP classevery
- A in NP is polynomial time reducible to B

If B is NP-complete and $B \leq_P C for C$ in NP, then C is NP-complete.

Proof — We know that C is in NP, so we need to show that every A in NP is polynomial time reducible to C to satisfy the second condition.

Because B is NP-complete, every language in NP is polynomial time reducible to B and B is polynomial time reducible to C.

However, polynomial time reductions have the property of composition. Thus, if A is polynomial time reducible to B and B is polynomial time reducible to C, A is polynomial time reducible to C. Thus, the second condition is satisfied for C and thus, it is NP-complete.

The Vertex Cover Problem

If *G* is an undirected graph, a vertex cover of *G* is a subset of the nodes where every node of *G* touches one of those nodes.
The problem asks whether a graph contains a vertex cover of specified size:

VERTEX−COVER={<*G*,*k*> ∣ *G is an undirected graph that has a k−node vertex cover*}.

Vertex Cover is NP-complete

Proof:

We need to reduce from 3SAT to Vertex-Cover to prove it. This reduction happens in polynomial time.The reduction maps a Boolean formula φ to G and a value k. For each variable x in φ, we produce an edge connecting two nodes and label them x and $\bar{x}$ where selecting the node x for the vertex cover means setting x to be TRUE and selecting the node $\bar{x}$ for the vertex cover means setting x to be FALSE.

The gadgets for the clauses are a bit more complex. Each clause gadget is a triple of nodes that are labeled with the three literals of the clause. These three nodes are connected to each other and to the nodes in the variable gadgets that have the identical labels. Thus, total number of nodes in G are

$2m + 3l$ where φ has m variables and l clauses. Let k be $m + 2l$. For example, if

$\varphi = (x1 \vee x1 \vee x2) \wedge (\bar{x}1 \vee \bar{x}2 \vee \bar{x}2) \wedge (\bar{x}1 \vee x2 \vee x2)$,

the reduction produces $< G, k >$ from φ, where $k = 2 + (2 \ast 3) = 8$.

Now, to prove that this reduction works, we need to show that φ **is satisfiable iff** G **has a vertex cover with** k **nodes**. We start with a satisfying argument:

1. Put the nodes of the variable gadgets that correspond to the true literals in the assignment into the vertex cover.
2. Select one true literal in every clause and put the remaining two nodes from every clause gadget into the vertex cover. This gives us k nodes covering all edges (every variable gadget edge is clearly covered, all three edges within every clause gadget are covered and all edges between variable and clause gadgets are covered). Thus, G has a vertex cover with k nodes.

Next, we show that if G has a vertex cover with k nodes, φ is satisfiable by constructing the satisfying assignment. The vertex cover must contain one node in each variable gadget and two in every clause gadget in order to cover the edges of the variable gadgets and the three edges within the clause gadgets. This will ensure that all nodes are accounted for.

We take the nodes of the variable gadgets that are in the vertex cover and assign TRUE to the corresponding literals. That assignment satisfies φ because each of the three edges connecting the variable gadgets with each clause gadget is covered, and only two nodes of the clause gadget are in the vertex cover. This means that one of the edges must be covered by a node from a variable gadget and thus, we have made a satisfying assignment.

SUBSET-SUM

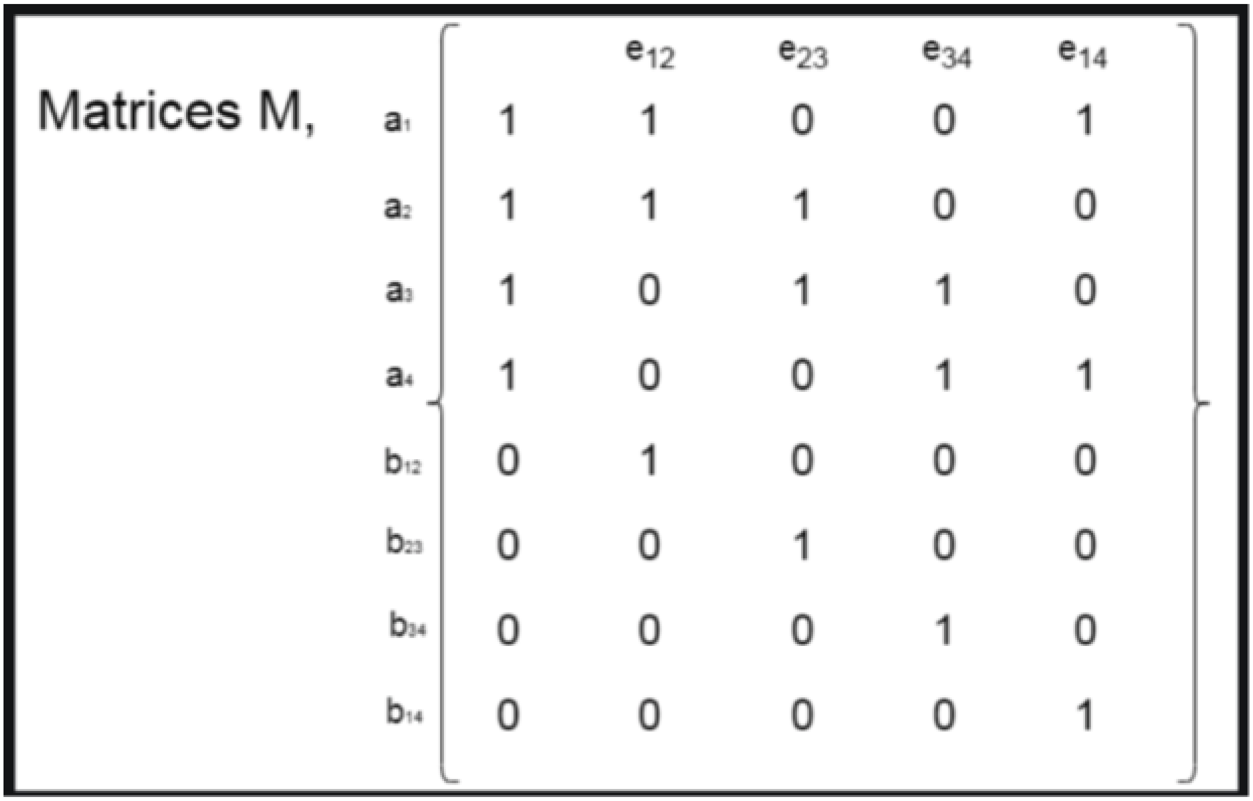
Determine if there is a subset of the provided set with sum equal to the supplied sum given a set of non-negative integers and a value sum.

Susbset Sum is NP-Complete. The issue is finding a subset of size "k" that sums to "sum."

- **Choose a problem `Y` that is known to be P-Complete in N:**

Problem 'Y' is the vertex cover problem.

All we need to show is that `Y` has a solution if and only if `Z` has a solution
'Y': Assume `Vertex Cover` to be defined as `G = (V, E), k`
The transformation: `a_i = i` for each vertex `i`. `a_i` is the set of vertices adjacent to `i`. `a_i` is a subset of `V`. `a_i` is a vertex cover if and only if it is a subset of `V` and each vertex in `a_i` is adjacent to at least one other vertex in `a_i`. We define a component called `b_ij` for each edge (i, j) .
The integers will be represented in a matrix format, with each row representing the corresponding integer value in 'base-4' format of '|E|+1' digits. As seen in the diagram below:



Matrices M,

$$\begin{array}{c|ccccc}
 & & e_{12} & e_{23} & e_{34} & e_{14} \\
a_1 & 1 & 1 & 0 & 0 & 1 \\
a_2 & 1 & 1 & 1 & 0 & 0 \\
a_3 & 1 & 0 & 1 & 1 & 0 \\
a_4 & 1 & 0 & 0 & 1 & 1 \\
b_{12} & 0 & 1 & 0 & 0 & 0 \\
b_{23} & 0 & 0 & 1 & 0 & 0 \\
b_{34} & 0 & 0 & 0 & 1 & 0 \\
b_{14} & 0 & 0 & 0 & 0 & 1
\end{array}$$

Points to note:

- Column 1 has `a_i = 1` and `b_{ij} = 0`
- If an edge exists, the column `(i,j) = 1`
- $d = k(4^{|E|}) + \sum^{|E| - 1}_{i = 0} 2(4^i)$