Reduction of IND to Subset_Sum

We define an instance of IND, the independent set problem, to be a string of the form $\langle G\rangle\langle k\rangle$ where G is a graph and k a positive integer. That string is a member of the language IND if there is some set I vertices of G such that $|I| = k$ and no two members of I form an edge of G.

We define an instance of *Subset_Sum*, the *subset sum problem* to consist of a number C, and a list of positive numbers $X = x_1, \dots x_p$. The ordered pair (C, X) is a member of the language *Subset_Sum* if there is some set of numbers S in the range $\{1, \dots p\}$ such that $\sum \{x_i : i \in S\} = C$.

We first prove that Subset_Sum is $\mathcal{NP}$, using the certificate method. If $(C, X) \in$Subset_Sum, then the set S itself is a certificate, and verification program is straightforward: simply add the weights.

We define a $\mathcal{P}$-TIME reduction R of IND to Subset_Sum, where IND is the independent set problem.

We assume that all of our languages (problems) are over $\Sigma = \{0, 1\}$, the binary alphabet. and each reduction must be a function

$$R : \Sigma^* \to \Sigma^*$$

. When we define $R(w)$ below, we will assume that w is an instance of the independent set problem. If w is any other string, we define $R(w) = \varepsilon$, the empty string. No further discussion of this case is necessary.

Let $G = (V, E)$ be a graph, and k a number. Then $\langle G\rangle\langle k\rangle$ is an instance of IND, the independent set problem. We define $R(\langle G\rangle\langle k\rangle)$, an instance of the subset sum problem, as follows.

Write $V = \{v_1, \dots v_n\}$ and $E = \{e_1, \dots e_m\}$, the vertices and edges of G, respectively. We say that v_i *meets* e_i if v_i is one of the two end points of e_j.

We will have two classes of entries in our list X, those derived from vertices of G, and those derived from edges of G. We call these $Y = y_1, \dots y_n$ and $z_1 \dots z_m$. We will then let $p = n + m$, and $X = Y + Z$, the concatenation of the two lists. More formally:

- For any $1 \le i \le n$, let $y_i = 10^{m+1} + \sum \left\{10^j : v_i \text{ meets } e_j\right\}$.
- For any $1 \le j \le m$, let $z_j = 10^j$
- For any $1 \le \ell \le n + m$, we define $x_\ell = \begin{cases} y_\ell \text{ if } \ell \le n \\ z_{n+\ell} \text{ otherwise} \end{cases}$ Then, let $X = x_1, \dots x_{n+m}$.
- Let $C = k\,10^{m+1} + \sum_{j=1}^{m} 10^j$

By the following two lemmas, R is a reduction of IND to Subset_Sum.

Two Lemmas?

Lemma 1 *If G has an independent set of size k, then $(C, X) \in$ Subset_Sum.*

Proof: Suppose $\mathcal{I}$ is a set consisting of k vertices of G. Let $\mathcal{J}$ be the set of edges which do not meet any of the vertices in $\mathcal{I}$. Define

- $Y = \{y_i : v_i \in \mathcal{I}\}$

- $Z = \{z_j : e_j \in \mathcal{J}\}$
- $X = Y \cup Z$

We claim that $\sum X = C$. We analyze the sum by examining "places," just like we did in elementary school.

We first examine places 1 through m, namely the coefficients of 10^j for each j. If there is some vertex v_i that meets e_j, y_i contributes 1 to that place. Since $\mathcal{I}$ has at most one vertex which meets e_j, and since $e_j \notin \mathcal{J}$, the coefficient of X in the j^{th} place is 1, just as in C. On the other hand, if there is no vertex that meets e_j, then there is no y_i which contributes to the j^{th} place of $\sum X$, but z_j does contribute a 1 in that place. In either case, $\sum X$ and C agree in the j^{th} place.

Finally, we note that, disregarding those first m places, $\sum X$ has k copies of 10^{m+1}, as does C, and we are done. ∎

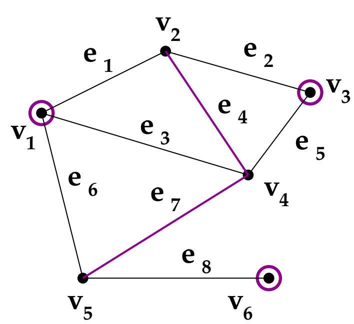
Trivially, R is a polynomial time function. Thus IND$\subseteq_{\mathcal{P}}$Subset_Sum.

Theorem 1 *If* IND *is* $\mathcal{NP}$–COMPLETE *then* Subset Sum *is* $\mathcal{NP}$–COMPLETE.

Example

Let G be the graph illustrated below, where $n = 6$ and $m = 8$. Let $k = 3$. The set $\mathcal{I} = \{v_1, v_3, v_6\}$ is an independent set of vertices of G of size k. In our reduction, $\mathcal{J} = \{e_4, e_7\}$. We write k and all the weights in base 10. The first array shows the weights of all items, while the second array shows that the weights of the selected items sum to k.

C	=	3	1	1	1	1	1	1	1	1	0
y_1	=	1	0	0	1	0	0	1	0	1	0
y_2	=	1	0	0	0	0	1	0	1	1	0
y_3	=	1	0	0	0	1	0	0	1	0	0
y_4	=	1	0	1	0	1	1	1	0	0	0
y_5	=	1	1	1	1	0	0	0	0	0	0
y_6	=	1	1	0	0	0	0	0	0	0	0
z_1	=	0	0	0	0	0	0	0	0	1	0
z_2	=	0	0	0	0	0	0	0	1	0	0
z_3	=	0	0	0	0	0	0	1	0	0	0
z_4	=	0	0	0	0	0	1	0	0	0	0
z_5	=	0	0	0	0	1	0	0	0	0	0
z_6	=	0	0	0	1	0	0	0	0	0	0
z_7	=	0	0	1	0	0	0	0	0	0	0
z_8	=	0	1	0	0	0	0	0	0	0	0



x_1	=	1	0	0	1	0	0	1	0	1	0
x_3	=	1	0	0	0	1	0	0	1	0	0
x_6	=	1	1	0	0	0	0	0	0	0	0
y_4	=	0	0	0	0	0	1	0	0	0	0
y_7	=	0	0	1	0	0	0	0	0	0	0
C	=	3	1	1	1	1	1	1	1	1	0