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 an indenpendent set I of vertices of G of cardinality k, that is, no two members of I form an edge of G.

We define an instance of *Subset_Sum*, the *subset sum problem* to be a string $\langle X\rangle\langle K\rangle$ where X is a list of positive integers and K is an integer. A *solution* to that instance is a sublist of X whose total is K. We have already given a $\mathcal{P}$–TIME reduction of 3-SAT to IND, and a $\mathcal{P}$–TIME reduction of SAT to 3-SAT. By the Cook-Levin theorem, SAT is $\mathcal{NP}$-complete, and therefore IND is $\mathcal{NP}$–complete. We prove that Subset_Sum is $\mathcal{NP}$, using the certificate method. The sublist whose total equals K is the certificate, which can be (trivially) verified in polynomial time. We define a $\mathcal{P}$-TIME reduction R of IND to Subset_Sum, proving that Subset_Sum is $\mathcal{NP}$–complete.

Definition and Verification of the Reduction. Let $\langle G\rangle\langle k\rangle$ be an instance of IND, where $G = (V, E)$. Write $V = \{v_0, v_1, v_2, \ldots v_{n-1}\}$, the vertices of G, and $E = \{e_0, e_1, e_2, \ldots e_m\}$, the edges of G. We say e_j *meets* v_i, and write $e_j \perp v_i$, if v_i is one of the two end points of e_j.

In our proof, we write all integers in base $B = \max(4, n+1)$. This will prevent "carrys" when we add weights. We define *weights* of each vertex and each edge: $W(v_i) = B^m + \sum_{e_j \perp v_i} B^j$, and $W(e_j) = B^j$.

Let $X = W(v_0), \ldots W(v_{n-1}), W(e_0), \ldots W(e_{m-1}))$, the list of weights of the vertices and the edges of G. Then $\sum X = nB^m + 3\sum_{j=0}^{m-1} B^j$, and no carrys will be needed. Let $K = kB^m + \sum_{j=0}^{m-1} B^j$. We define $R(\langle G\rangle\langle k\rangle) = \langle X\rangle\langle K\rangle$, an instance of the subset sum problem.

Theorem 1 *G has an independent set of cardinality k if and only if X has a sublist whose sum is K.*

Proof: Suppose I is an independent set of k vertices of G. Let J be the set of all edges which do not meet any member of I, and let $W = \sum_{v\in I} W(v) + \sum_{e\in J} W(e)$.

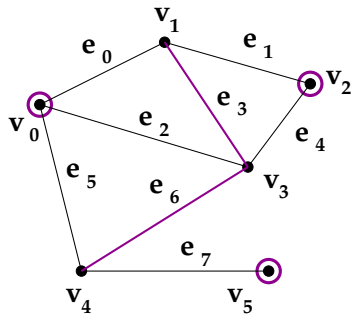
Claim: $W = K$. Since $|I| = k$, the weights of vertices contribute k to the m^{th} digit of W (numbering digits from right to left), while the weights of edges contribute nothing to that digit. Thus, the m^{th} digits of W and K agree. For $0 \le j < n$, The j^{th} digit of K is 1. If $e_j \in J$, then it contributes 1 to the j^{th} digit of W, and vertices contributed nothing. On the other hand, if e_j meets a member of I, say v_i, it is not a member of J, hence contributeds nothing to the j^{th} digit of W, while v_i contributes 1. Since I is independent, the vertex at the other end of e_j also contributes nothing to that digit. This proves the claim, hence $\langle X\rangle\langle K\rangle$ has a solution if $\langle G\rangle k$ has a solution.

Conversely, Suppose that some sublist of X has total K. Since there are no carrys, and the m^{th} digit of W is k, W must be the sum of exactly k weights of vertices, plus perhaps some weights of edges. Let I be the set of those vertices. We claim that I is independent, since otherwise two of them would have an edge e_j in common, making the the j^{th} digit of W at least 2, contradiction. ∎

Example

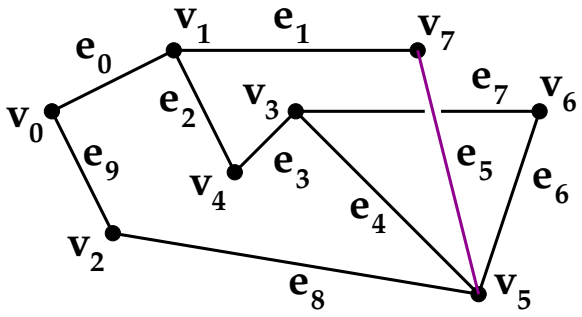
Let G be the graph illustrated below, where $n = 6$ and $m = 8$. Let $k = 3$. The set $I = \{v_1, v_3, v_6\}$ is an independent set of vertices of G of size k. In our reduction, $J = \{e_4, e_7\}$. We write K and all the weights in base $B = 7$. The first array shows the weights of all items, while the second array shows that the weights of the selected items sum to K.

		B^8	B^7	B^6	B^5	B^4	B^3	B^2	B^1	B^0
y_0	=	1	0	0	1	0	0	1	0	1
y_1	=	1	0	0	0	0	1	0	1	1
y_2	=	1	0	0	0	1	0	0	1	0
y_3	=	1	0	1	0	1	1	1	0	0
y_4	=	1	1	1	1	0	0	0	0	0
y_5	=	1	1	0	0	0	0	0	0	0
z_0	=	0	0	0	0	0	0	0	0	1
z_1	=	0	0	0	0	0	0	0	1	0
z_2	=	0	0	0	0	0	0	1	0	0
z_3	=	0	0	0	0	0	1	0	0	0
z_4	=	0	0	0	0	1	0	0	0	0
z_5	=	0	0	0	1	0	0	0	0	0
z_6	=	0	0	1	0	0	0	0	0	0
z_7	=	0	1	0	0	0	0	0	0	0
$\sum W$	=	6	3	3	3	3	3	3	3	3



		B^8	B^7	B^6	B^5	B^4	B^3	B^2	B^1	B^0
y_0	=	1	0	0	1	0	0	1	0	1
y_2	=	1	0	0	0	1	0	0	1	0
y_5	=	1	1	0	0	0	0	0	0	0
z_3	=	0	0	0	0	0	1	0	0	0
z_6	=	0	0	1	0	0	0	0	0	0
K	=	3	1	1	1	1	1	1	1	1

Exercise 1: Let G be the graph shown here. Then $\langle G \rangle \langle 4 \rangle$ is an instance of IND. Compute $R(\langle G \rangle \langle 4 \rangle)$, an instance of Subset_Sum. Is there are solution?



Pseudopolynomial Algorithm for Subset Sum

We define an instance of *Subset_Sum*, the *subset sum problem* to be a string $\langle X\rangle\langle K\rangle$ where $X = x_1, x_2, \ldots x_n.$ is a list of positive integers and K is an integer. A *solution* to that instance is a subsequence of X whose total is K. Without loss of generality, $x_i \leq K$ for each i, since otherwise x_i could not be part of the solution.

We have already proved that Subset_Sum is $\mathcal{NP}$ complete. We now give a dynamic programming algorithm $\mathcal{A}$ of time complexity $O(nK)$.

Definition of $\mathcal{A}$. Let $A[n+1][K+1]$ be the Boolean matrix where $A[i][k]$ means there is a subsequence of $x_1, \ldots x_i$ whose sum is k.

For k from 1 to K
    $A[0][k] = 0$
A[0][0] = 1
For i from 1 to n
    For k from 0 to K
        A[i][k] = A[i-1][k]
        If $k \geq x_i$ and $A[i][k - x_i]$
            A[i][k] = 1
Return $A[n][K]$

Exercise 2: Use the algorithm $\mathcal{A}$ to decide whether there is a sublist of (6, 10, 7, 17, 3, 7, 10, 3, 4) whose sum is 25. (In a sequence, there can be duplicate terms.)

Exercise 3: Why can't we call $\mathcal{A}$ "polynomial"?

Exercise 4: However, $\mathcal{A}$ can be made $\mathcal{P}$–TIME, if the terms of the sequence are restricted to numbers with at most 2 digits. Explain.