2. ***When both minimizing and maximizing are easy …*** This does not mean that deciding the existence of a solution in a given interval is easy. Recall that the problem of computing a MINIMUM SPANNING TREE can be solved by a polynomial time greedy algorithm. The same strategy also works for the MAXIMUM SPANNING TREE, so both the *extremal* versions of the spanning tree problem are in P. The following EXACT SPANNING TREE problem asks for a spanning tree whose weight is in a given interval, somewhere between these extremes:

   Input: A weighted graph G and two integers l and u.

   Question: Is there a spanning tree T with weight w such that $l \leq w \leq u$?

   It's not clear how to generalize the greedy algorithm to EXACT SPANNING TREE. In fact, there is probably no polynomial time algorithm at all, because EXACT SPANNING TREE is NP-complete. Show that EXACT SPANNING TREE is NP-complete.

Para demostrar que EXACT SPANNING TREE es NP-Completo, basta con demostrar que existe un problema L que también es NP-Completo que podemos reducir a EXACT SPANNING TREE, es decir L ≤ EXACT SPANNING TREE.

La reducción la haremos desde desde el problema NP-Completo SUBSET SUM, por lo tanto procedemos a demostrar que existe una reducción válida y computable en tiempo polinómico SUBSET SUM ≤ EXACT SPANNING TREE.

SUBSET SUM consiste en que dado un conjunto S formado por enteros, y un entero W, determinar si existe un subconjunto K de S el cual la suma de sus elementos es igual a W.

Dada una entrada de SUBSET SUM cualquiera, la reduciremos de la siguiente manera: crearemos un Grafo estrella G con n +1 vértices donde |S| = n; el vértice central se conectará al resto de los n vértices con aristas de peso correspondientes a cada uno de los elementos de S. El intervalo l ≤ w ≤ u haremos que sea W = l = u. Cada vértice exterior lo uniremos a los dos que tiene a los lados con aristas de peso 0.

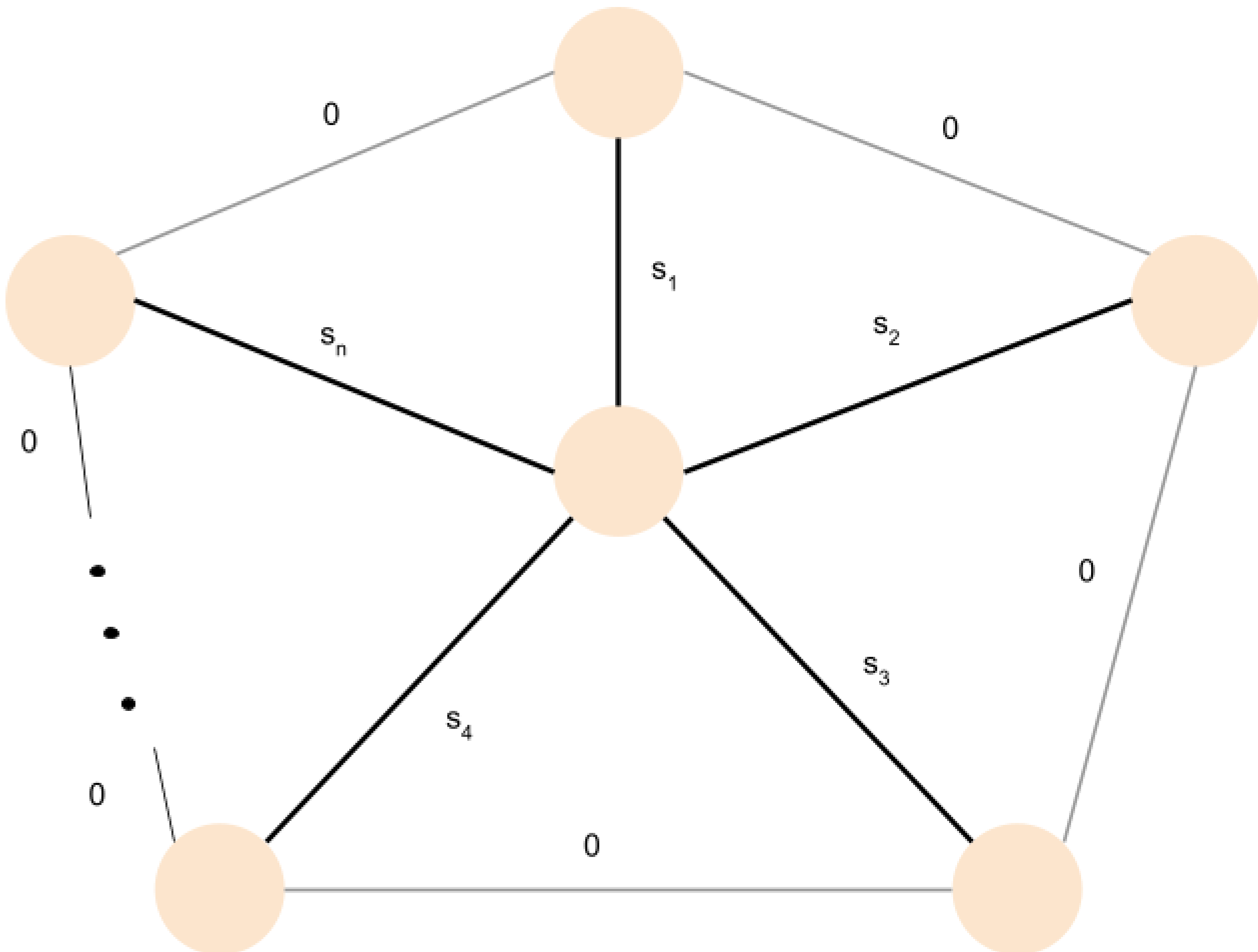
Crear un grafo con n +1 vértices se puede hacer en tiempo polinómico respecto a la entrada, siendo esta el numero de elementos del conjunto S. Añadir una arista desde el vértice central a cada uno de los vértices exteriores también se puede computar en tiempo polinómico respecto a la entrada, ya que solo hay que añadir una arista por cada elemento del conjunto S.
Añadir a cada vértice exterior dos aristas que lo unan con los dos vértices exteriores que se encuentren al lado también se puede hacer en tiempo polinómico respecto a la entrada.

Por lo tanto nuestra reducción es computable en tiempo polinómico respecto a la entrada del problema SUBSET SUM.

Si la entrada de SUBSET SUM es positiva, entonces también existirá un spanning tree de peso W, que consistirá en coger en G las aristas del vértice central al resto de vértices con un peso equivalente a las del subconjunto K el cual sus elementos suman W. El resto de vértices que no han sido utilizados (las aristas incidentes al vértice central con pesos S \ K) formarán parte del spanning tree mediante las aristas de peso 0 que le unen con los vértices que ya forman parte del spanning tree.

Si la entrada de SUBSET SUM es negativa, entonces no existirá un spanning tree de peso W, y por lo tanto se preservara la entrada negativa en EXACT SPANNING TREE.

Con esto hemos demostrado que nuestra reducción es correcta, ya que la entrada de EXACT SPANNING TREE será positiva si y sólo si también es positiva para SUBSET SUM.



$S = \{s_i | \ 1 \leq i \ \leq n : |S| = n\}$