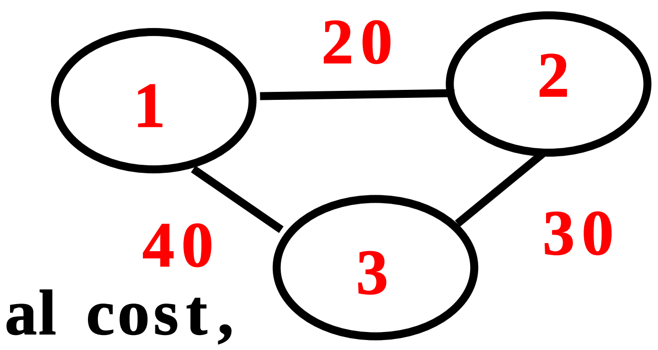
Connecting Stars and Planets

Elon Musk has tasked you with connecting a network of stars and planets using a special material called G10.

Each possible connection between a pair of nodes (stars or planets) has an associated cost, which may be negative, zero, or positive.

Your mission is to ensure that all stars and planets are fully connected, either directly or indirectly.



In other words, the entire solar system is connected

You must find a set of connections that achieves the minimum possible total cost, defined as the sum of the selected connection costs and you must also report the individual cost of each selected connection.

min_sum=50
min-prod=600
ws=[20,30]
wp=[20,30]

Special requirement:

If the weights of all possible connections are greater than 0,
Elon wants a configuration of connections that optimizes for the absolute minimum product of the connection costs, and you must also report
the individual cost of each selected connection.

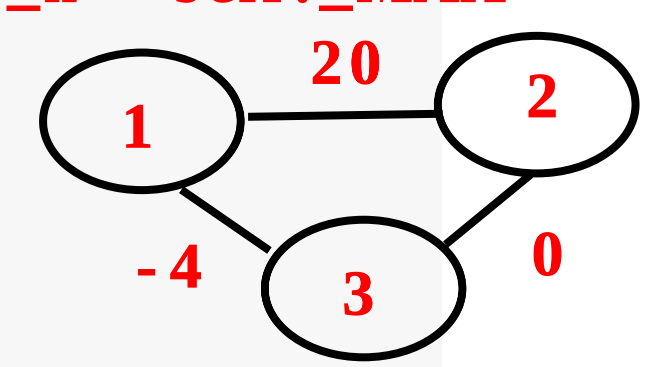
LIMIT=self._MAX = 100

#DO NOT BUILD GRAPH or COMPUTE PRODUCT if self._n > self._MAX

```
class GraphExam:
  def __init__(
        self,
        n:"num verices",
        g: "graph",
        graph_name: "string",
        a: "python list of tuples (from, to, weight)",
        min_sum:'list of size 1', #Must compute always
        min_prod:'list of size 1', #compute only if all weights > 0 and n <= LIMIT
        ws:'List of all path weights used for computing min MST sum',
        wp:'List of all path weights used for computing min MST product',
        show:'bool' #Must show SUM graph if show = true
```

Must build and work on graph if n <100

Will work on a if n>100



min_sum= -4
min_prod=-1
ws=[0,-4]
wp=[]

Must print input graph and output graph with weights