1. Hamiltonian Path to TSP
   - Hamiltonian path
     ![Graph of Hamiltonian path](image)
   - TSP
     ![Graph of TSP](image)
   - there is an edge between every pair of vertices.
   - threshold = 4
   - reduction
     ![Reduction Diagram](image)
     - total length = 3
     - total length 5

$\mathcal{L} = n - 1$
   - in the intended solution
     - $n-1$ edges in the original graph cost 1
     - 1 edge $(s, t)$

Step 2: easy Hamiltonian path from $s$ to $t$
   - add edge $(s, t)$ to form a cycle for TSP
Step 3: if there is a cycle in TSP with total length \( n-1 \) n edges there is only 1 edge (s,t) length 0 all other edges have length \( \geq 1 \) only way to have total length = n-1 is to include (s,t), and only use edges of length 1.

3. Tripartite matching

0 Tripartite matching to Subset Vectors

select a subset of hyperedges select a subset of vectors

map hyperedges \( \rightarrow \) vectors
vertices \( \rightarrow \) component of vectors.

3 n vertices dimension of vectors 3n

hyper \((1, 1, 2)\) \(\uparrow \uparrow \uparrow \) \(\rightarrow \) \(\uparrow \rightarrow \) \(\uparrow \)

select n hyper-edges s.t. each vertex is in exactly 1 hyperedge
2) subset vector → Subset sum

select vectors  select number

idea 1: think of vectors as binary numbers.

not correct because sum of vectors different from sum of numbers

\((0, 0, 1, 0) + (0, 1, 1, 0) = (0, 1, 2, 0)\)

\((010)_b + (0110)_b = (1000)_b\)

idea 2: carry operations in sum

use a number with a larger basis

so a carry operation cannot happen

\((0, 0, 1, 0) \leq (0, 1, 1, 0)\)

\((0010)_k + (0110)_k = (0120)_k\)

just need to choose base \(k\) to be large enough.

\(k \geq n\) is enough for \(n\) binary vectors.