TSP (graph)

Input: a complete weight graph

output: Hamitonian cycle with the minimum weight and the minimum weight

---

Let mst be getMST(graph)

Let dfs_path be getDFSPath(mst)

Let Hamitonian_cycle be getHamitonianCycle(dfs_path)

Let weight be getWeight(Hamitonian_cycle)

return weight, Hamitonian cycle

get MST (graph):

input: a complete weighted graph

output: a minimum spanning tree

---

Let S be an empty set

Let Q be an empty priority queue

Let mst_edges be an empty list

Add root node "0" to queue

while queue is not empty, do:

    Let current be node with min weight

    if current not in S:

        Add current to S

        Add current to mst_edge

    for all neighbor of current:

        if neighbor not in S

            Append neighbor to queue.

make it a traversable tree

return toTree(mst_edges)

get DFS Path ( mst, root )

inputs: minimum spanning tree, a root node

output: Depth First search path

---

```
let path be an empty list
let neighbors be neighbors of root
if neighbors is empty:
        return

for all neighbor in neighbors:
    append to path: getDFSPath(mst, neighbor)
    append root to path

return path
```

get Hamiltonian Cycle ( dfs_path )

input: depth first search path of a tree

output: Hamiltonian cycle represented by a list of vertex

---

Let result be an empty list

for all vertex in dfs_path:

    if vertex is not in empty list:

        append vertex to result

append first node dfs_path[0] to result

return result

get Hamiltonian Weight ( cycle )

input: List of vertex representing Hamiltonian cycle

output: Total weight of cycle

---

Let a = 0

Let b = 1

Let total_weight be 0

while b less than length of cycle

Let weight be weight between vertex at index a and index b

total_weight = total_weight + weight

a += 1

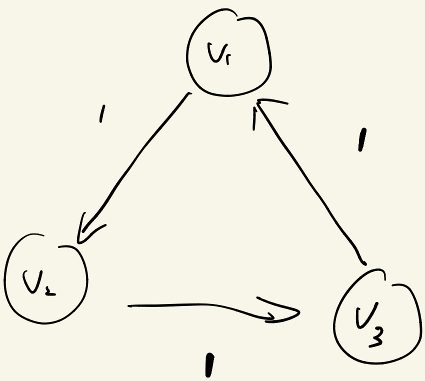
b += 1

return weight

## Proof:

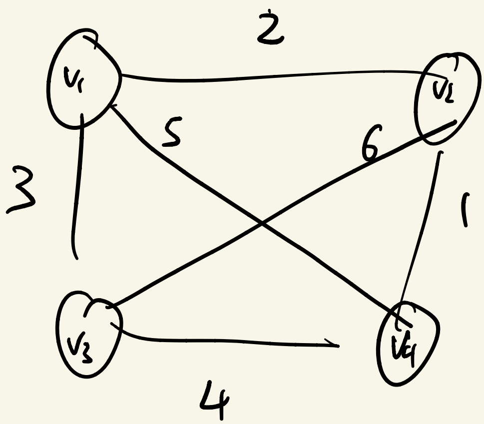
**Lemma:** There exist a complete, metrically weighted graph $G = (V, E)$ such that your approximation always finds a Hamiltonian cycle of min weight.

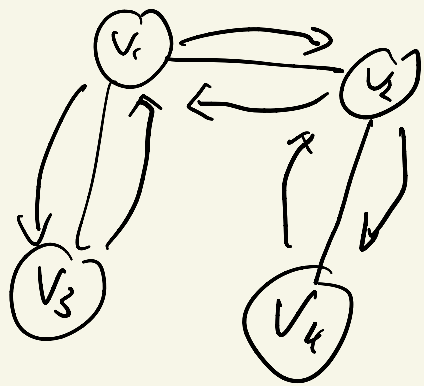
Example:




TSP will get a min MST with $\{(v_1, v_2), (v_2, v_3)\}$ and the dfs will be $[v_1, v_2, v_3, v_2, v_1]$ and path is $[v_1, v_2, v_3]$ so weight is 3 always

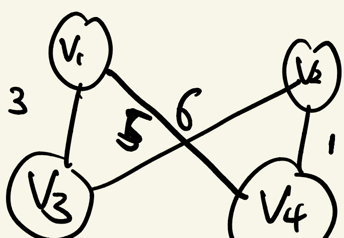
**Lemma:** There exist a complete, metrically weighted graph $G = (V, E)$ such that your approximation may find a Hamiltonian cycle of greater than minimum weight.

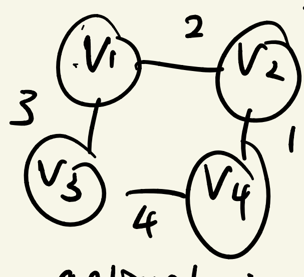



dfs = $[\, v_1, v_3, v_1, v_2, v_4, v_2, v_1 \,]$

Hamiltonian path: $[\, v_1, v_3, v_2, v_4 \,]$




TPS returns

$1+3+5+6 = \boxed{15}$

Optimal is

$1+2+3+4 = \boxed{10}$