

Congratulations! You passed!

Grade received 100%

Latest Submission Grade 100%

To pass 80% or higher

Go to next item

2-Approximation

Quiz • 2h 2m

Review Learning Objectives

1.

Implement an approximation algorithm for the Traveling Salesman problem, which outputs a solution which is a 2-approximation of the optimal weight.

1 / 1 point

Submit your assignment

Due Jan 15, 11:59 PM IST

1

import networkx as nx

2

3

This function takes as input a graph g.

4

The graph is complete (i.e., each pair of distinct vertices is connected by an edge),

5

undirected (i.e., the edge from u to v has the same weight as the edge from v to u),

6

and has no self-loops (i.e., there are no edges from i to i).

7

#

8

The function should return a 2-approximation of an optimal Hamiltonian cycle.

9

10

def approximation(g):

11

n is the number of vertices.

12

n = g.number_of_nodes()

13

14

You might want to use the function "nx.minimum_spanning_tree(g)"

15

which returns a Minimum Spanning Tree of the graph g

16

17

You also might want to use the command "list(nx.dfs_preorder_nodes(graph, 0))"

18

which gives a list of vertices of the given graph in depth-first preorder.

19

20

cycle = list(nx.dfs_preorder_nodes(nx.minimum_spanning_tree(g), 0)) + [0]

21

22

return sum(g[cycle[i]][cycle[i+1]]['weight'] for i in range(len(cycle[: -1])))

23

Run

Reset

Like

Dislike

Report an issue

Good job!

