Item Navigation

Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

Nearest Neighbors

Quiz • 30 min

Review Learning Objectives

1. Implement the Nearest Neighbors Heuristic for the Traveling Salesman Problem. Your algorithm should start with the vertex number 0, and then each time select the closest vertex among the ones which don't yet belong to the cycle. **1 / 1 point**

Submit your assignment

Due Jan 1, 11:59 PM IST

Try again

Receive grade

To Pass 80% or higher

Your grade

100%

View Feedback

We keep your highest score

Like Dislike Report an issue

```
import networkx as nx

# This function takes as input a graph g.
# The graph is complete (i.e., each pair of distinct vertices is connected by an edge),
# undirected (i.e., the edge from u to v has the same weight as the edge from v to u),
# and has no self-loops (i.e., there are no edges from i to i).
#
# The function should return the weight of the nearest neighbor heuristic, which starts at the vertex number 0,
# and then each time selects a closest vertex.


def nearest_neighbors(g):
    current_node = 0
    path = [current_node]
    n = g.number_of_nodes()

    # We'll repeat the same routine (n-1) times
    for _ in range(n - 1):
        next_node = None
        # The distance to the closest vertex. Initialized with infinity.
        min_edge = float("inf")
        for v in g.nodes():
            if not v in path:
                if g[current_node][v]['weight'] < min_edge:
                    min_edge, next_node = g[current_node][v]['weight'], v
        # Write your code here: decide if v is a better candidate than next_node.
        # If it is, then update the values of next_node and min_edge

        assert next_node is not None
        path.append(next_node)
        current_node = next_node

    weight = sum(g[path[i]][path[i + 1]]['weight'] for i in range(g.number_of_nodes() - 1))
    weight += g[path[-1]][path[0]]['weight']
    return weight

# You might want to copy your solution to your Jupiter Notebook to see how close this heuristic is to the optimal solu
```

Run

Reset

Correct

Good job!