Traveling Salesman Problem

Approximation presentation

Pseudocode for the approximation

```
Def nearest_v_tsp(graph, n):
    Visited = [False] *n
    Path[0]
    Total = 0
    For _ in range (n)
        Cur_v = path[-1]
        Nearest = None
        min_dist = float('inf')
        For v in range (n):
            If v not visited and graph[cur_v][v] < min_dist:
                Nearest = v
                Min_dist = graph[cur_v][v]
        path.append(nearest)
        visited[nearest) = true
        Total += min_dist
```

Greedy pseudocode for approximation

The TSP greedy algorithm is being greedy with the shortest path to any connected vertices with no regard to the total length of the whole traversal. The hope is that by consistently choosing the nearest vertices that the algorithm will end up with a solution that is reasonably close to the optimal route.

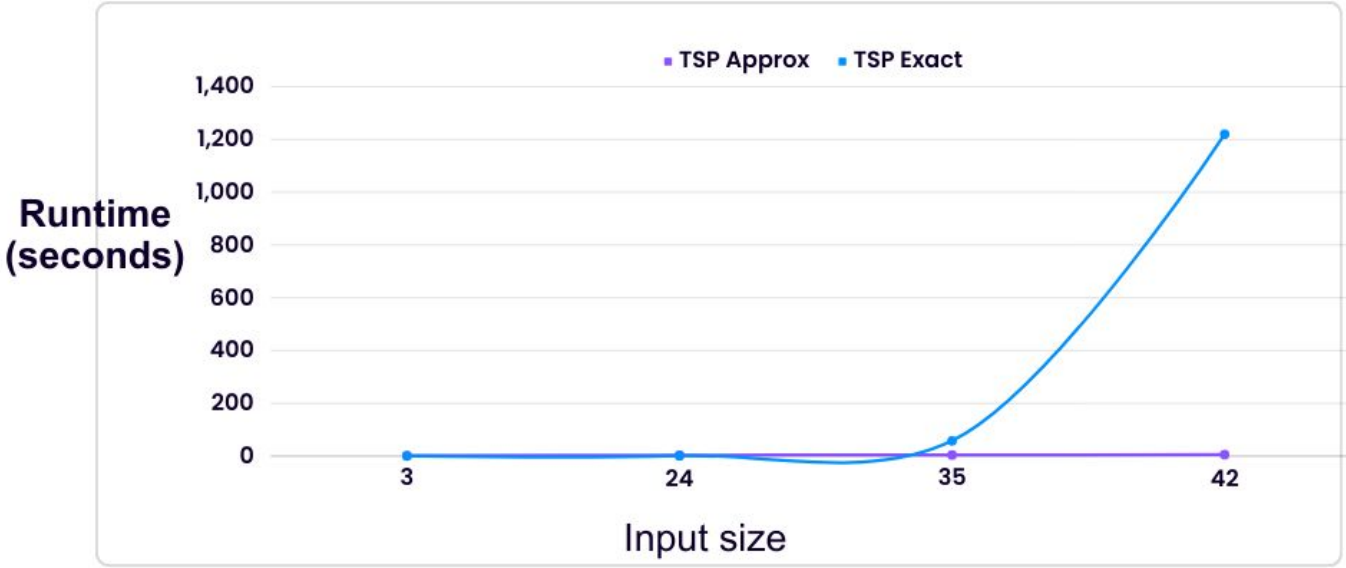
The Worse Case O Time

```
For _ in range (n)
        Cur_v = path[-1]
        Nearest = None
        min_dist = float('inf')
        For v in range (n):
                If v not visited and graph[cur_v][v] < min_dist:
                        Nearest = v
                        Min_dist = graph[cur_v][v]
        path.append(nearest)
        visited[nearest) = true
        Total += min_dist
```

O(n *(n-1)) = O(n^2)

Plot illustration

TSP Runtime



Exact vs Approximation