**AI Assignment 3**

**Neeti Kurulkar**

**Greedy Algorithms**

1. **Prim’s Algorithm for Minimal Spanning Tree**

**Code:**

**class** **Graph:**

**def** \_\_init\_\_**(**self**,** v**):**

self**.***v* **=** v

self**.***adj\_mat* **=** **[[**0 **for** \_ **in** **range(**v**)]** **for** \_ **in** **range(**v**)]**

**def** add\_edge**(**self**,** a**,** b**,** w**):**

self**.***adj\_mat***[**a**][**b**]** **=** w

self**.***adj\_mat***[**b**][**a**]** **=** w

**def** prim\_mst**(**self**,** start**):**

visited **=** **[False]** **\*** self**.***v*

visited**[**start**]** **=** **True**

**print(**"\nEdge : Weight"**)**

edge\_count **=** 0

**while** edge\_count **<** self**.***v* **-** 1**:**

minimum **=** **float(**'inf'**)**

x**,** y **=** **-**1**,** **-**1

**for** i **in** **range(**self**.***v***):**

**if** visited**[**i**]:**

**for** j **in** **range(**self**.***v***):**

**if** **not** visited**[**j**]** **and** self**.***adj\_mat***[**i**][**j**]** **!=** 0**:**

**if** minimum **>** self**.***adj\_mat***[**i**][**j**]:**

minimum **=** self**.***adj\_mat***[**i**][**j**]**

x**,** y **=** i**,** j

**print(**f"{x}-{y}: {minimum}"**)**

visited**[**y**]** **=** **True**

edge\_count **+=** 1

n **=** **int(input(**"Enter number of vertices: "**))**

g **=** Graph**(**n**)**

e **=** **int(input(**"Enter number of edges: "**))**

**print(**"Enter edges (u v w):"**)**

**for** \_ **in** **range(**e**):**

u**,** v**,** w **=** **map(int,** **input().***split***())**

g**.***add\_edge***(**u**,** v**,** w**)**

start **=** **int(input(**"\nEnter starting vertex for MST: "**))**

g**.***prim\_mst***(**start**)**

**Output:**

A screen shot of a computer

Description automatically generated

1. **Selection Sort Algorithm**

**Code:**

**def** get\_input**():**

n **=** **int(input(**"Enter the number of elements in the array: "**))**

arr **=** **[]**

**for** i **in** **range(**n**):**

element **=** **int(input(**f"Enter element {i**+**1}: "**))**

arr**.***append***(**element**)**

**return** arr

**def** selection\_sort**(**arr**):**

"""Selection Sort Algorithm"""

**for** i **in** **range(len(**arr**)):**

min\_idx **=** i

**for** j **in** **range(**i **+** 1**,** **len(**arr**)):**

**if** arr**[**j**]** **<** arr**[**min\_idx**]:**

min\_idx **=** j

arr**[**i**],** arr**[**min\_idx**]** **=** arr**[**min\_idx**],** arr**[**i**]**

arr **=** get\_input**()**

**print(**"Original array:"**,** arr**)**

selection\_sort**(**arr**)**

**print(**"Sorted array:"**,** arr**)**

**Output:**

**A screen shot of a computer

Description automatically generated**