Name: Ashutosh Raj Gupta

Roll no: TECO2223A045

Batch: A3

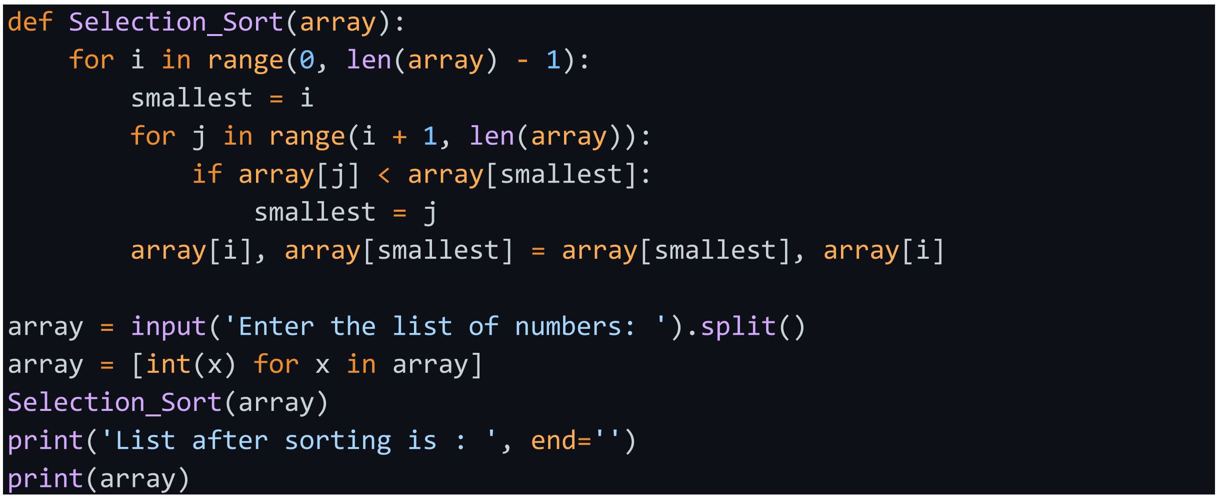
Assignment no: 3

Implement Greedy search algorithm for any of the following application:

I. Selection Sort
II. II. Minimum Spanning Tree
III. III. Single-Source Shortest Path Problem
IV. IV. Job Scheduling Problem
V. V. Prim's Minimal Spanning Tree Algorithm
VI. VI. Kruskal's Minimal Spanning Tree Algorithm
VII. VII. Dijkstra's Minimal Spanning Tree Algorithm

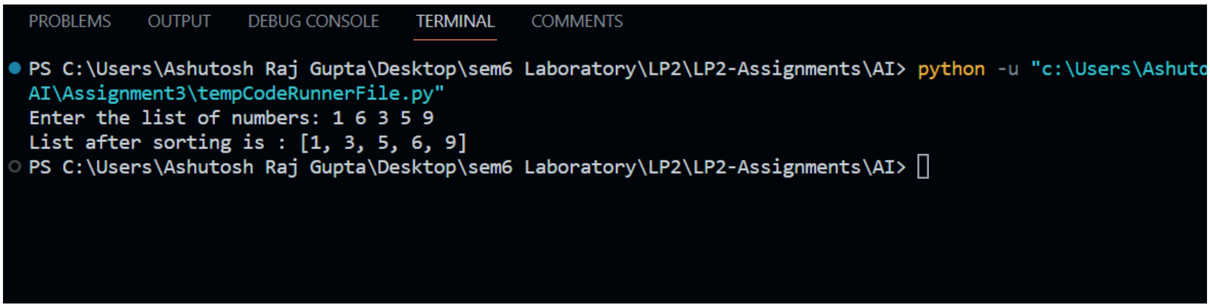
Code:

1.Selection sort:



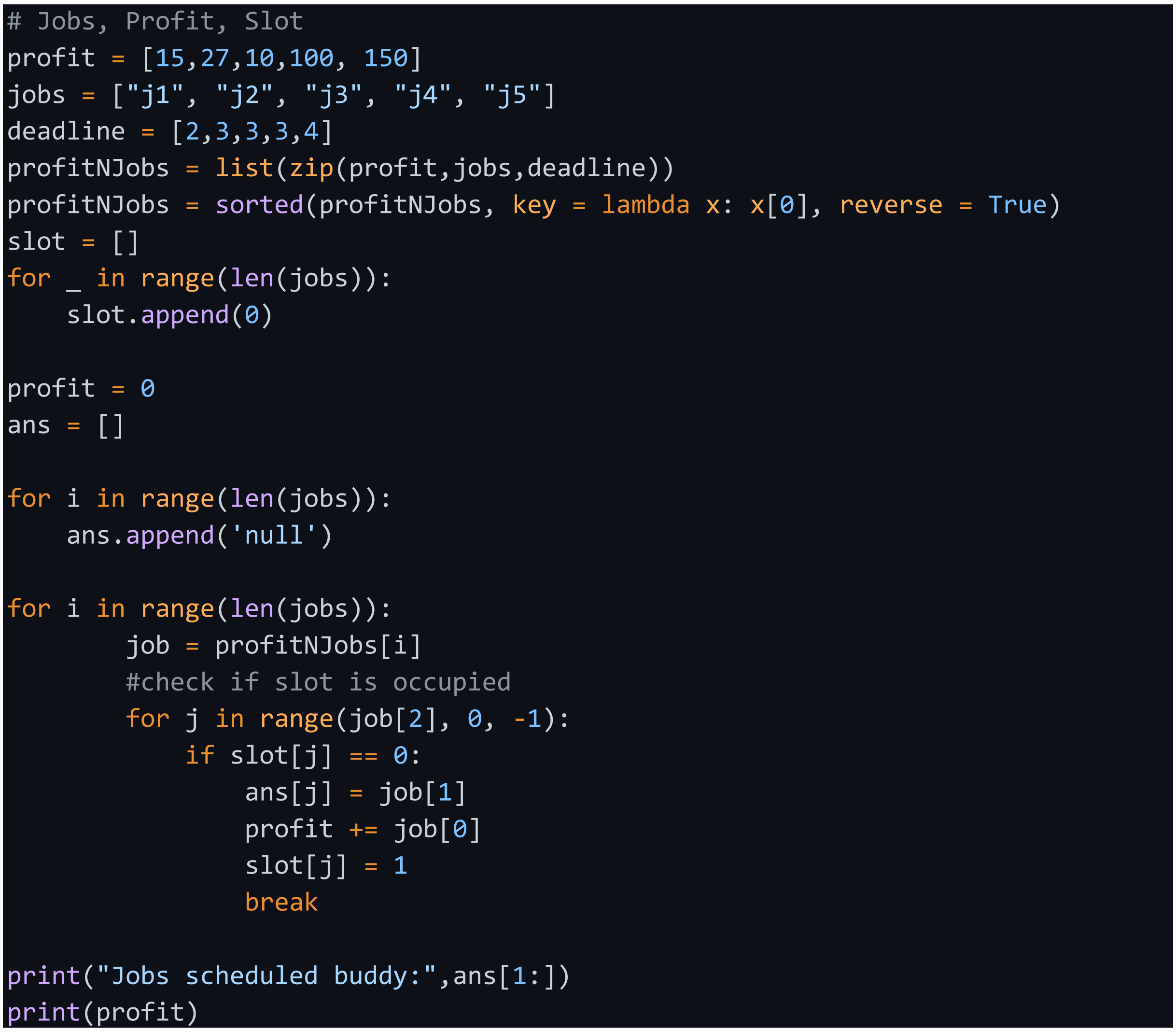
```
def Selection_Sort(array):
    for i in range(0, len(array) - 1):
        smallest = i
        for j in range(i + 1, len(array)):
            if array[j] < array[smallest]:
                smallest = j
        array[i], array[smallest] = array[smallest], array[i]

array = input('Enter the list of numbers: ').split()
array = [int(x) for x in array]
Selection_Sort(array)
print('List after sorting is : ', end='')
print(array)
```

output:



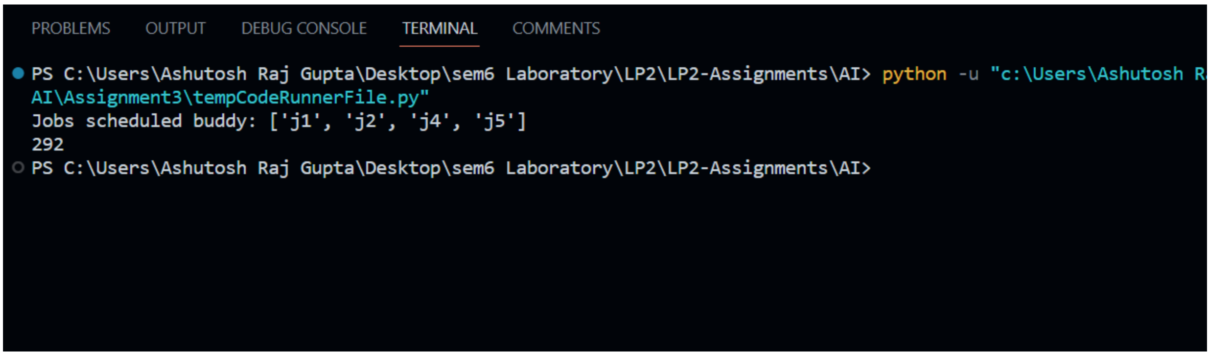
```
PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI> python -u "c:\Users\Ashuto
AI\Assignment3\tempCodeRunnerFile.py"
Enter the list of numbers: 1 6 3 5 9
List after sorting is : [1, 3, 5, 6, 9]
PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI>
```

Code 2: Job scheduling



```
# Jobs, Profit, Slot
profit = [15,27,10,100, 150]
jobs = ["j1", "j2", "j3", "j4", "j5"]
deadline = [2,3,3,3,4]
profitNJobs = list(zip(profit,jobs,deadline))
profitNJobs = sorted(profitNJobs, key = lambda x: x[0], reverse = True)
slot = []
for _ in range(len(jobs)):
    slot.append(0)

profit = 0
ans = []

for i in range(len(jobs)):
    ans.append('null')

for i in range(len(jobs)):
        job = profitNJobs[i]
        #check if slot is occupied
        for j in range(job[2], 0, -1):
            if slot[j] == 0:
                ans[j] = job[1]
                profit += job[0]
                slot[j] = 1
                break

print("Jobs scheduled buddy:",ans[1:])
print(profit)
```

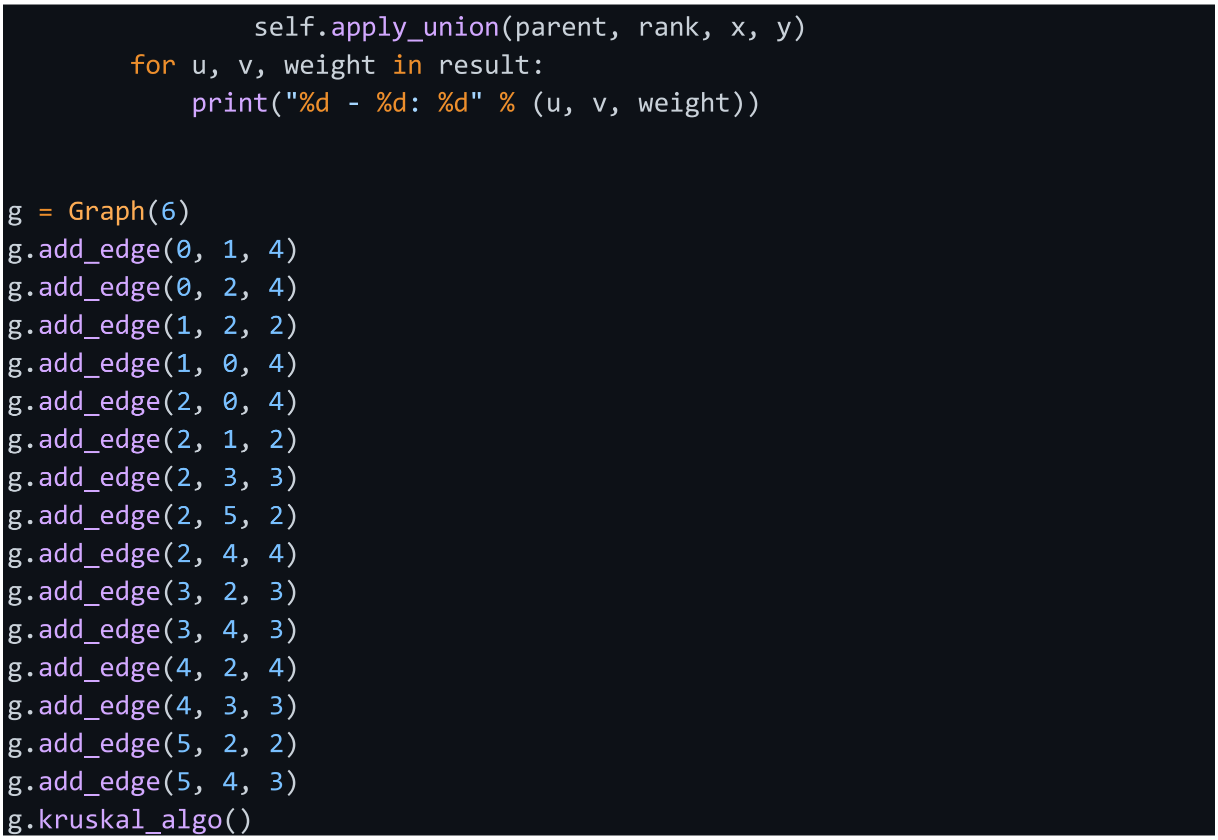
output:



```
● PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI> python -u "c:\Users\Ashutosh R
  AI\Assignment3\tempCodeRunnerFile.py"
  Jobs scheduled buddy: ['j1', 'j2', 'j4', 'j5']
  292
○ PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI>
```

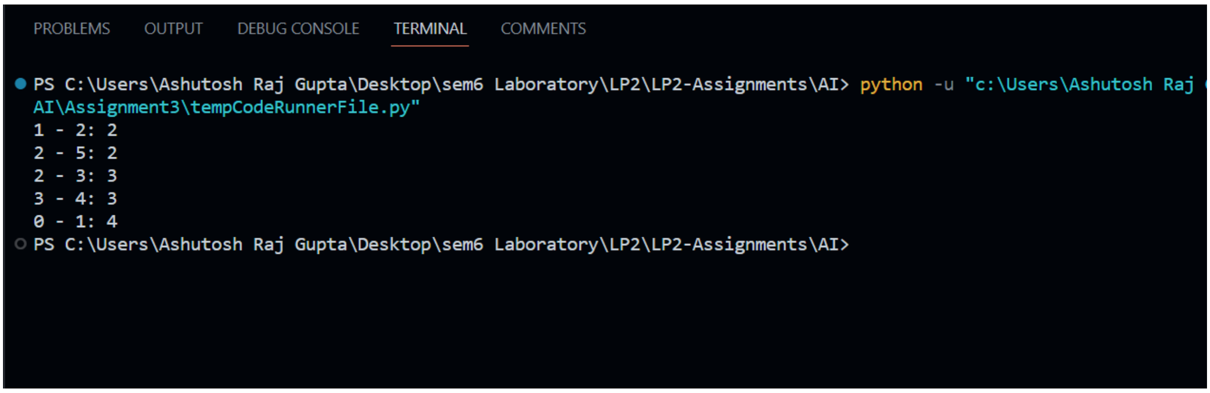
Code 3: Kruskals

```
# Kruskal's algorithm in Python


class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = []

    def add_edge(self, u, v, w):
        self.graph.append([u, v, w])

    # Search function

    def find(self, parent, i):
        if parent[i] == i:
            return i
        return self.find(parent, parent[i])

    def apply_union(self, parent, rank, x, y):
        xroot = self.find(parent, x)
        yroot = self.find(parent, y)
        if rank[xroot] < rank[yroot]:
            parent[xroot] = yroot
        elif rank[xroot] > rank[yroot]:
            parent[yroot] = xroot
        else:
            parent[yroot] = xroot
            rank[xroot] += 1

    #  Applying Kruskal algorithm
    def kruskal_algo(self):
        result = []
        i, e = 0, 0
        self.graph = sorted(self.graph, key=lambda item: item[2])
        parent = []
        rank = []
        for node in range(self.V):
            parent.append(node)
            rank.append(0)
        while e < self.V - 1:
            u, v, w = self.graph[i]
            i = i + 1
            x = self.find(parent, u)
            y = self.find(parent, v)
            if x != y:
                e = e + 1
                result.append([u, v, w])
```



```
                self.apply_union(parent, rank, x, y)
        for u, v, weight in result:
            print("%d - %d: %d" % (u, v, weight))


g = Graph(6)
g.add_edge(0, 1, 4)
g.add_edge(0, 2, 4)
g.add_edge(1, 2, 2)
g.add_edge(1, 0, 4)
g.add_edge(2, 0, 4)
g.add_edge(2, 1, 2)
g.add_edge(2, 3, 3)
g.add_edge(2, 5, 2)
g.add_edge(2, 4, 4)
g.add_edge(3, 2, 3)
g.add_edge(3, 4, 3)
g.add_edge(4, 2, 4)
g.add_edge(4, 3, 3)
g.add_edge(5, 2, 2)
g.add_edge(5, 4, 3)
g.kruskal_algo()
```

output-



```
● PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI> python -u "c:\Users\Ashutosh Raj 
  AI\Assignment3\tempCodeRunnerFile.py"
  1 - 2: 2
  2 - 5: 2
  2 - 3: 3
  3 - 4: 3
  0 - 1: 4
○ PS C:\Users\Ashutosh Raj Gupta\Desktop\sem6 Laboratory\LP2\LP2-Assignments\AI>
```