

Name:- Kaveri Sandip Gaikwad

Roll no:- C03016

ASSIGNMENT NO:-03

Implement Greedy search algorithm for any of the following application:

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

```
import sys, heapq
```

```
from collections import defaultdict
```

```
from math import inf
```

```
def selectionSort(A):
```

```
    U = A.copy()
```

```
    for i in range(len(A)):
```

```
        min_idx = i
```

```
        for j in range(i+1, len(A)):
```

```
            if A[min_idx] > A[j]:
```

```
                min_idx = j
```

```
        A[i], A[min_idx] = A[min_idx], A[i]
```

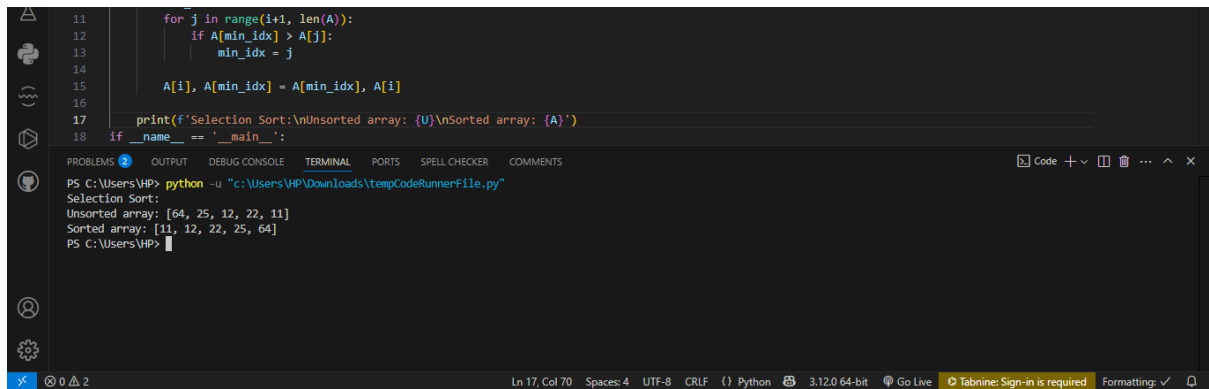
```
    print(f'Selection Sort:\nUnsorted array: {U}\nSorted array: {A}')
```

```
if __name__ == '__main__':
```

```
    A = [64, 25, 12, 22, 11]
```

```
    selectionSort(A)
```

OUTPUT:



The image shows a Visual Studio Code editor window with a Python script implementing Selection Sort. The code is as follows:

```
11     for j in range(i+1, len(A)):
12         if A[min_idx] > A[j]:
13             min_idx = j
14
15     A[i], A[min_idx] = A[min_idx], A[i]
16
17     print(f'Selection Sort:\nUnsorted array: {U}\nSorted array: {A}')
18 if __name__ == '__main__':
```

The terminal output shows the execution of the script:

```
PS C:\Users\HP> python -u "c:\Users\HP\Downloads\tempCodeRunnerFile.py"
Selection Sort:
Unsorted array: [64, 25, 12, 22, 11]
Sorted array: [11, 12, 22, 25, 64]
PS C:\Users\HP>
```

The status bar at the bottom indicates the file is at Line 17, Column 70, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.12.0 64-bit. It also shows a Go Live button and a Tabnine sign-in notification.