19CSE302 Design and Analysis of Algorithms

Lab Sheet 6

S Abhishek

AM.EN.U4CSE19147

Colab

BFS of Directed Graph

```
graph = {

  '5' : ['3','7'],
  '3' : ['2', '4'],
  '7' : ['8'],
  '2' : [],
  '4' : ['8'],
  '8' : []

}

v = []
q = []

def BFS(v, graph, node):

  v.append(node)

  q.append(node)

  while q:

    m = q.pop(0)

    print (m, end = " ")

    for neighbour in graph[m]:
```

```
if neighbour not in v:

        v.append(neighbour)

        q.append(neighbour)

if __name__ == "__main__":

  print("BFS : ",end = " ")

  BFS(v, graph, '5')
```

```
BFS :  5 3 7 2 4 8
```

BFS of Undirected Graph

```
graph = {

  '5' : ['3','7'],
  '3' : ['2', '4', '5'],
  '7' : ['5', '8'],
  '2' : ['3'],
  '4' : ['3', '8'],
  '8' : ['4', '7']

}

v = []
q = []

def BFS(v, graph, node):

  v.append(node)

  q.append(node)

  while q:

    m = q.pop(0)

    print (m, end = " ")
```

```
    for neighbour in graph[m]:

      if neighbour not in v:

        v.append(neighbour)

        q.append(neighbour)

if __name__ == "__main__":

  print("BFS : ",end = " ")

  BFS(v, graph, '5')
```

```
BFS :  5 3 7 2 4 8
```

DFS of Directed Graph

```
graph = {

  '5' : ['3','7'],
  '3' : ['2', '4'],
  '7' : ['8'],
  '2' : [],
  '4' : ['8'],
  '8' : []

}

v = set()

def DFS(visited, graph, node):

    if node not in v:

        print(node, end = " ")

        v.add(node)

        for neighbour in graph[node]:
            DFS(v, graph, neighbour)
```

```
if __name__ == "__main__":

  print("DFS : ",end = " ")

  DFS(visited, graph,'5')
```

```
DFS :  5 3 2 4 8 7
```

DFS of Undirected Graph

```
graph = {

  '5' : ['3','7'],
  '3' : ['2', '4', '5'],
  '7' : ['5', '8'],
  '2' : ['3'],
  '4' : ['3', '8'],
  '8' : ['4', '7']

}

v = set()

def DFS(visited, graph, node):

    if node not in v:

        print(node, end = " ")

        v.add(node)

        for neighbour in graph[node]:

            DFS(v, graph, neighbour)


if __name__ == "__main__":

  print("DFS : ",end = " ")

  DFS(visited, graph,'5')
```

```
DFS :  5 3 2 4 8 7
```

Trace the Node

```
graph = {

  '5' : ['3','7'],
  '3' : ['2', '4'],
  '7' : ['8'],
  '2' : [],
  '4' : ['8'],
  '8' : []

}

v = []
q = []

def BFS(v, graph, node):

  v = []
  q = []

  v.append(node)

  q.append(node)

  while q:

    m = q.pop(0)

    for neighbour in graph[m]:

      if neighbour not in v:

        v.append(neighbour)

        q.append(neighbour)

  v.pop(0)

  print(f"When Source is {node} : {v}")
```

```
if __name__ == "__main__":

  for i in list(graph.keys()):

    BFS(v, graph, i)
```

```
When Source is 5 : ['3', '7', '2', '4', '8']
When Source is 3 : ['2', '4', '8']
When Source is 7 : ['8']
When Source is 2 : []
When Source is 4 : ['8']
When Source is 8 : []
```

Shortest Path Visiting All Nodes

Submission Detail

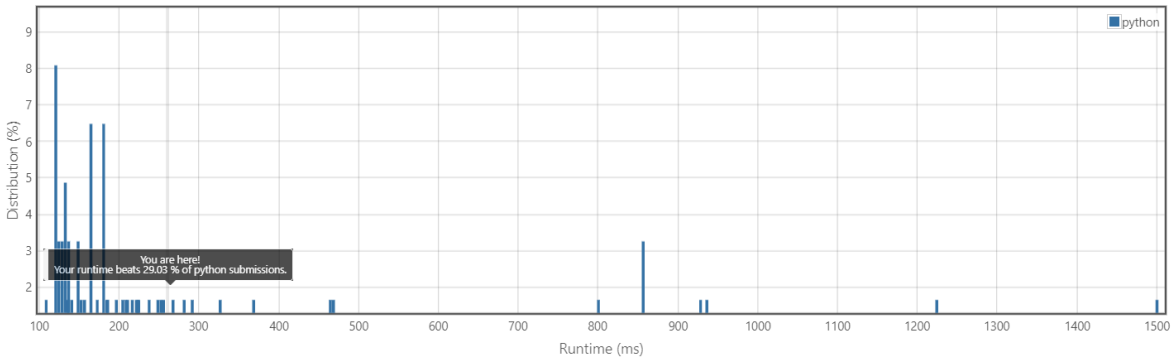
51 / 51 test cases passed.

Runtime: **261 ms**
Memory Usage: **14.6 MB**

Status: **Accepted**

Submitted: **3 minutes ago**

Accepted Solutions Runtime Distribution



Cheapest Flights Within K Stops

Submission Detail

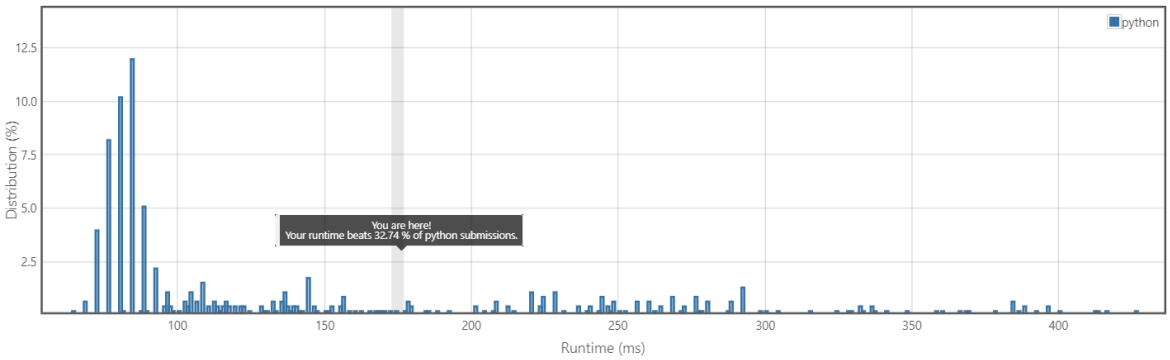
49 / 49 test cases passed.

Runtime: **175 ms**
Memory Usage: **15.3 MB**

Status: **Accepted**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Course Schedule

Submission Detail

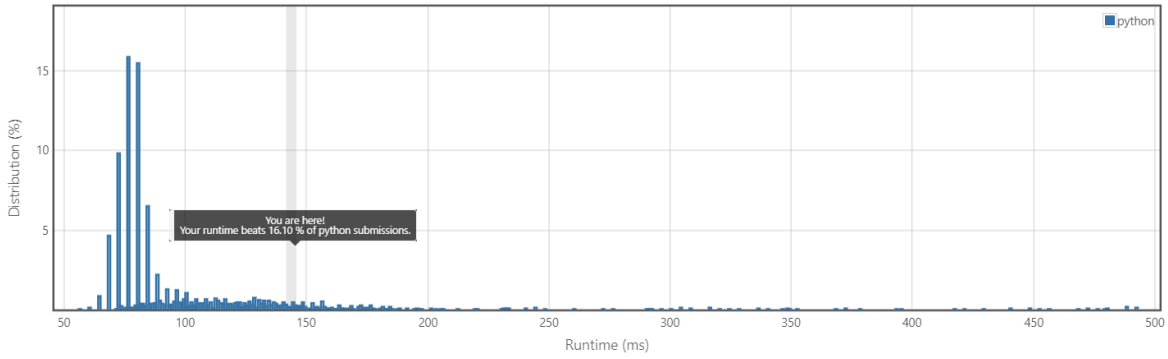
51 / 51 test cases passed.

Runtime: **144 ms**
Memory Usage: **15.5 MB**

Status: **Accepted**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Course Schedule IV

Submission Detail

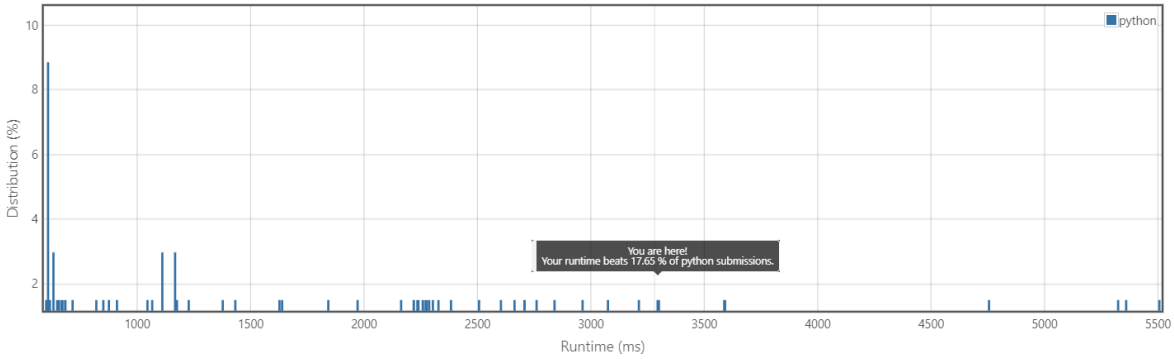
68 / 68 test cases passed.

Runtime: **3286 ms**
Memory Usage: **17.2 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Clone Graph

Submission Detail

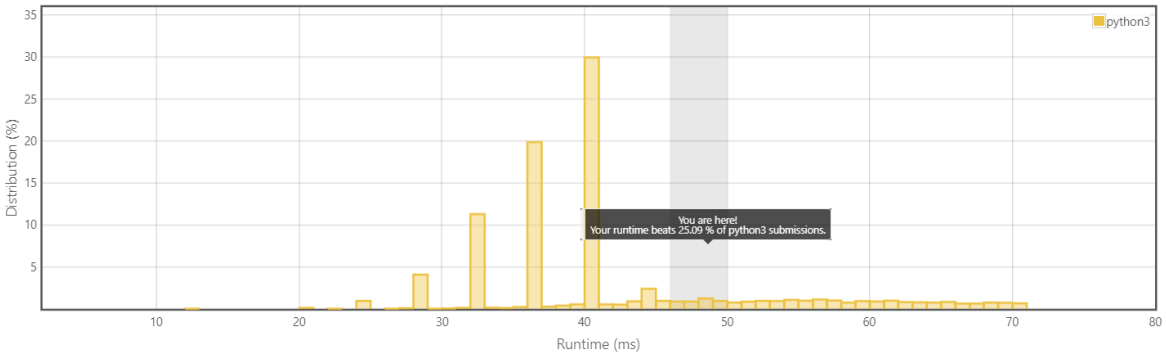
22 / 22 test cases passed.

Runtime: **48 ms**
Memory Usage: **14.5 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Thankyou!!