+ Code + Text

```
import pandas as pd
df=pd.read_csv("/content/6citytsp.csv",header=None).values
print(df)
```

```
[[  0  64 378 519 434 200]
 [ 64   0 318 455 375 164]
 [378 318   0 170 265 344]
 [519 455 170   0 223 428]
 [434 375 265 223   0 273]
 [200 164 344 428 273   0]]
```

```
from itertools import permutations
citynames=list(range(df.shape[0]))
per=permutations(citynames)
per=list(per)
print(type(per))
print(per)
```

```
<class 'list'>
[(0, 1, 2, 3, 4, 5), (0, 1, 2, 3, 5, 4), (0, 1, 2, 4, 3, 5), (0, 1, 2, 4, 5, 3), (0, 1, 2, 5, 3, 4), (0, 1, 2, 5, 4, 3), (0, 1, 3,
```

```
#BRUTE - FORCE APPROACH

import numpy as np
import time as tm
st=tm.process_time()
besttourlength=np.inf
for i in per:
  tourlength=0
  for j in range(len(i)-1):
    tourlength+=df[i[j],i[j+1]]
  tourlength+=df[i[j+1],i[0]]
  if tourlength<besttourlength:
    besttourlength=tourlength
    besttour=i
et=tm.process_time()
print("Time :- ",(et-st)*1000)
print("Best Tour :- ",besttour)
print("Best Tour Length :- ",besttourlength)
```

```
Time :-  3.4531829999995267
Best Tour :-  (0, 1, 2, 3, 4, 5)
Best Tour Length :-  1248
```

```
#BRUTE - FORCE WITH USER INPUT

import numpy as np
import time as tm
startcity=int(input("Enter a start city :- "))
citynames=list(range(df.shape[0]))
citynames.remove(startcity)
st=tm.process_time()
besttourlength=np.inf
per=list(permutations(citynames))
for i in per:
  tourlength=0
  for j in range(len(i)-1):
    tourlength+=df[i[j],i[j+1]]
  tourlength+=df[i[j+1],startcity]
  tourlength+=df[startcity,i[0]]
  if tourlength<besttourlength:
    besttourlength=tourlength
    besttour=i
besttour=list(besttour)
besttour.insert(0,startcity)
et=tm.process_time()
print("Time :- ",(et-st)*1000)
print("Best Tour :- ",besttour)
print("Best Tour Length :- ",besttourlength)
```

```
Enter a start city :- 0
Time :-  0.8644440000011855
Best Tour :-  [0, 1, 2, 3, 4, 5]
Best Tour Length :-  1248
```

```
#NEAREST NEIGHBOUR HEURISTIC APPROACH

import pandas as pd
import numpy as np
```

```
import numpy as np
import time as tm

df=pd.read_csv('/content/6citytsp.csv',header=None).values.astype(float)
startcity=int(input("Enter a start city :- "))
citynames=list(range(df.shape[0]))
st=tm.process_time()

tourlength=0
besttour=[startcity]
df[df==0]=np.inf
df1=df.copy()

for i in range(df.shape[0]-1):
  if i==0:
    tourlength+=min(df[startcity,:])
    nextbestind=np.argmin(df[startcity,:])
    besttour.append(nextbestind)
    df[:,nextbestind]=np.inf
    df[:,startcity]=np.inf
  else:
    tourlength+=min(df[nextbestind,:])
    nextbestind=np.argmin(df[nextbestind,:])
    df[:,nextbestind]=np.inf
    besttour.append(nextbestind)

tourlength+=df1[nextbestind,startcity]
et=tm.process_time()

time=(et-st)*1000
print("BEST TOUR IS :- ",besttour)
print("BEST TOUR LENGTH IS :- ",tourlength)
print("TIME TAKEN :- ",time)
```

```
Enter a start city :- 0
BEST TOUR IS :-  [0, 1, 5, 4, 3, 2]
BEST TOUR LENGTH IS :-  1272.0
TIME TAKEN :-  0.7996749999996666
```