IE 343 TERM PROJECT

Furkan Baran S020525

Meriç Cesur S018111

For knapsack:

Due to an heuristic algorithm is wanted from us, we used an algorithm that sorts value divided by weight for each of the song. If an exact method is wanted we can use top down or bottom up algorithm .We used value as popularity and then duration of songs as weight . Here our reason to use that algorithm is time performance of it. Actually this algorithm can give optimal solution for some cases. After we sorted them by using their value/weight we start putting it to song list until we reach the concert duration. The complexity of algorithm is $O(nlogn)$ where there are n songs. Because it used sorted function is used timsort as an algorithm and it consumes very short time.

Here is the unit test of Knapsack for max duration equal to 200.

```
Song: Popularity - 69.0, Duration - 4.25955
Song: Popularity - 54.0, Duration - 3.5126666666666666
Song: Popularity - 53.0, Duration - 3.657766666666667
Song: Popularity - 42.0, Duration - 2.9351166666666666
Song: Popularity - 58.0, Duration - 4.108883333333333
Song: Popularity - 51.0, Duration - 3.9521333333333333
Song: Popularity - 48.0, Duration - 4.017766666666667
Song: Popularity - 30.0, Duration - 2.5544333333333333
Song: Popularity - 43.0, Duration - 3.67195
Song: Popularity - 50.0, Duration - 4.3691
Song: Popularity - 31.0, Duration - 2.786
Song: Popularity - 57.0, Duration - 5.147766666666667
Song: Popularity - 41.0, Duration - 3.7188833333333333
Song: Popularity - 34.0, Duration - 3.2840166666666666
Song: Popularity - 39.0, Duration - 3.81555
Song: Popularity - 39.0, Duration - 3.83355
Song: Popularity - 40.0, Duration - 3.96
Song: Popularity - 39.0, Duration - 3.93955
Song: Popularity - 31.0, Duration - 3.1326666666666667
Song: Popularity - 34.0, Duration - 3.752883333333333
Song: Popularity - 31.0, Duration - 3.4384333333333332
Song: Popularity - 47.0, Duration - 5.52155
Song: Popularity - 31.0, Duration - 3.745616666666667
Song: Popularity - 32.0, Duration - 3.898216666666667
Song: Popularity - 31.0, Duration - 3.8897666666666666
Song: Popularity - 32.0, Duration - 4.031916666666667
Song: Popularity - 33.0, Duration - 4.1887
Song: Popularity - 23.0, Duration - 2.970666666666667
Song: Popularity - 28.0, Duration - 3.6369666666666665
Song: Popularity - 29.0, Duration - 3.798966666666667
Song: Popularity - 42.0, Duration - 5.51195
Song: Popularity - 32.0, Duration - 4.21345
Song: Popularity - 30.0, Duration - 3.9971
Song: Popularity - 25.0, Duration - 3.3466666666666667
Song: Popularity - 42.0, Duration - 5.6386666666666665
Song: Popularity - 22.0, Duration - 2.9662166666666665
Song: Popularity - 28.0, Duration - 3.7891
Song: Popularity - 36.0, Duration - 5.08355
Song: Popularity - 28.0, Duration - 3.9837666666666665
Song: Popularity - 31.0, Duration - 4.464883333333334
Song: Popularity - 27.0, Duration - 3.894433333333333
Song: Popularity - 27.0, Duration - 3.9162166666666667
Song: Popularity - 34.0, Duration - 4.97755
Song: Popularity - 27.0, Duration - 3.9753333333333334
Song: Popularity - 24.0, Duration - 3.5537666666666667
Song: Popularity - 25.0, Duration - 3.7028833333333333
Song: Popularity - 26.0, Duration - 3.9026666666666667
Song: Popularity - 31.0, Duration - 4.65555
Song: Popularity - 30.0, Duration - 4.520883333333333
Song: Popularity - 25.0, Duration - 3.8148833333333334
```

For TSP

After that we used nearest neighborhood algorithm and this algorithm aim is find the closest place from current city by using Euclidean distance measure and then to prevent go back we change to distance with infinity after that we find path to travel around the determined concert places. This was an basic explanation of algorithm. Where there is n cities the complexity is $O(n^2)$ which is polynomial.

Here is the algorithm outcome for our city list.

```
Total Distance: 1037.1053121982652
Tour:

New York City
Washington D.C.
Ottawa
Mexico City
Los Angeles
Brasilia
Sao Paulo
Buenos Aires
Buenos Aires
Lagos
Nairobi
Addis Ababa
Riyadh
Abu Dhabi
Tehran
Karachi
Mumbai
Delhi
Islamabad
Bangkok
Hanoi
Manila
Seoul
Seoul
Osaka
Tokyo
Beijing
Beijing
Jakarta
Canberra
Wellington
Pretoria
Cairo
Cairo
Athens
Istanbul
Rome
Berlin
Oslo
Stockholm
Helsinki
Moscow
Moscow
Paris
Paris
London
Dublin
Madrid
Lisbon
```

For the final step we find the total popularity and total duration of concerts by given durations with applying knapsack algorithm for each of the concerts.

And the outcome is with total execution time is

```
Total Popularity Reach In Concerts: 37241.0
Total Duration In all of the Concerts: 3031.1690499999895
Execution Time is:0.00200653076171875
```