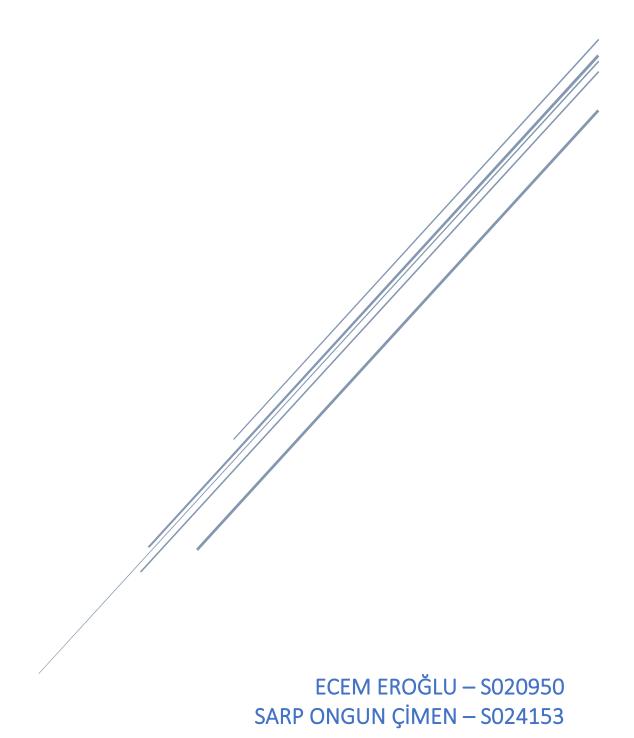
IE 343 TERM PROJECT



Introduction

This study aims to solve Knapsack and Traveling Salesman problems using Python

programming language. We used two data files named cities and song. The cities file contains

latitude, longitude, and concert duration data, and the song file contains song name,

popularity, and duration data. We have city and song classes to create their object instances.

Algorithmic Approach

We used the bottom-up algorithm to solve the Knapsack problem. We created a table with

one more row than the number of songs and one more column than the concert duration and

filled this table with zeros. We added one more row and column to account for the cases with

zero items and zero capacity. We used a nested loop consisting of two "for loops" to

implement the Knapsack algorithm. We applied the bottom-up algorithm by going through the

elements of the table one by one with this nested loop. We used a "while loop" to find the

songs we selected. The complexity of Knapsack is O(nW).

We used the Nearest Neighbor algorithm to solve the Travelling Salesman problem. In our

code, the cities we have been to are listed as True, and the cities we have not yet been listed as

False. At the beginning of the tour, all cities were False. We added the first selected city to the

tour, and then we completed the tour by going to the next closest city and visiting all cities.

The complexity of a Traveling Salesman is $O(n^2)$.

Output

Total distance: 972.2080791359843

Total popularity: 59260.0

Total duration: 2715.5037833333367

Execution time: 157.89270401000977 ms

Conclusion

We used the Knapsack algorithm for each city in the tour which we determined by using the

Nearest Neighbor algorithm, in order to find the optimal songs based on the duration of the

concert in that city. We calculated the total distance as 972, total popularity as 59260, total

duration as 2715, and execution time as 157 ms.