**Data Structure And Algorithm Project**

**Group Members:**

Ativ Joshi - 201501040  
Utsav Oza - 201501064  
Pratik Padalia - 201501084  
Suraj Patel - 201501105

**About the Project**

This project models an approach to find the most optimal path finder for NGOs to locate the unused food from restaurants and other places. This becomes a NP hard problem, thus a good approximation is analyzed and reported back minimizing calculations and maximizing efficiency heuristically.  
There are many instances when the restaurants want to give away all their leftover food but they do not find anyone to give it, while the NGOs who donate food amongst the needy do not find food easily and are not able to contact the restaurants and other places. This app will make it convenient for both of them to have the food not going into waste.  
We have used the approach of fractional knapsack with slight modifications where required. We have a fixed quantity (value), we need to minimize the distance (weight) having achieved the value. Different datasets were used to test the implementation. The simulation results has been successful achieved.

**Algorithm**

Pre-computation : (Clustering)

1. Fetch all nodes (food distributors : restaurants, etc) with their location.
2. Calculate distance of each node with every other node
3. Cluster the nearby nodes as per the threshold distance.

Main Algorithm : Fractional Knapsack (Modified)

1. Get the capacity of food that can be distributed by the NGO
2. Get NGO's current location
3. Fetch all clusters with their location and get the amount of food they can donate, value (sum of all nodes in that cluster)
4. Assign a weight to each cluster (Euclidian distance between our location and the cluster)
5. Sort the clusters in descending order of their ratio of value to weight
6. while required value >= value of the top most cluster in the list repeat steps 7,8 and 9
7. Select the cluster of the top of list
8. Decrease the required value equal to the value received from the cluster
9. Remove the cluster from the main list.
10. while required value > 0 repeat steps 11,12 and 13
11. Set the value of all other nodes = minimum of their current value and the required value
12. Sort the clusters again with their new ratio of value to weight
13. Select the nodes until the required value > 0 and keep decreasing the corresponding value simultaneously.
14. Display all the selected Nodes.

**Screenshot**

Screenshot 1 shows the list of restaurants selected from all the restaurants.

Screenshot 2 is the cluster of restaurants computed from the list of restaurants.

**Code**

**lib** folder contains the java files for core logic.

**lib\_clustering** contains the java files for clustering logic.

Note: the list of restaurants is read from Restaurant.json file stored in respective folders.

**Reference**

Introduction to Algorithms, Book by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen.

geeksforgeeks.com

https://www.youtube.com/user/**tusharroy**2525