README

- For question 2 the ingredients are picked randomly and not taken as input from the user.
- Copy mutate algorithm is implemented in the same way as mentioned in the research paper https://drive.google.com/file/d/1BRxheZTSHh1fHxj7ExUW_p21wt0WwMkI/view
- The dataset information can be seen with the help of data_info() function
- Total number of recipes generated after the end of all epochs = size of the dataset that is provdided (Kaggle Dataset)
- Complete information is displayed while running each epoch such as M ratio, Primordial cuisine size etc.
- Fitness value is assigneed randomly using predefined python function
- When the Kitchen Basket is expanding the Nature Basket is shrining
- To maintain that all the recipes present in the primordial cusine should be unique is done by making the recipes to be set. This will ensure that the recipe has no repititive ingredient as well as the primoridal cuisine will also contain all unique recipes
- Making the sets of recipes will also decrease the search time
- Time taken to run Q1 iand Q2 s noted to around 2 minutes.
- Each function name is self explanatory.
- freq_rank_plot(pc,epoch,choose): pc – primordial cusine (list of sets), epoch – epoch number , choose – ‘all’/’one’ where all means that it will plot all the graphs of epochs seprately where as for one means it’ll plot all the epochs in a single graph
- The same explanation goes for rec_size_distribution(pc,epoch,choose)
- copy_mutate_algorithm(pc,KB,epochs,total_size,NB,fitness,choose):
  pc - primordial cusine (list of sets)
  KB- Kitchen basket size (list)
  epochs - integer value
  total_size = 39774
  NB- nature basket
  fitness – dictionary of fitness value assigned
  choose – ‘all’/’one’ this is for graph
- The entire code is modular