**THE FOOD WORLD WANTS**



**GROUP MEMBERS:**

Suhana Rajput

Anmol Jumani

Ragini Gopchandani

**ORDER PROCESSING CODE:**

##Food ordering

lst=[] ---------------------------------------------------------------O(1)

total=[] -------------------------------------------------------------O(1)

def Order\_processing(Menu):

    price=0 ----------------------------------------------------------O(1)

    while True: ------------------------------------------------------O(n)

        keys=list(Menu.keys())----------------------------------------O(1)

        key=','.join(keys) -------------------------------------------O(1)

        print('Please select your cuisine among', str(key)) ----------O(1)

        cuisine = input()---------------------------------------------O(1)

        Dishes=Menu[cuisine] -----------------------------------------O(1)

        print('select your dishes from the chosen cuisine') ----------O(1)

        list\_of\_dishes = []-------------------------------------------O(1)

        dish = input()------------------------------------------------O(1)

        push(list\_of\_dishes, dish) -----------------------------------O(1)

        for i in list\_of\_dishes: -------------------------------------O(n)

            for j in i.split(','): -----------------------------------O(n)

                push(lst, j) ------------------------------O(1)

        for j in lst: ------------------------------------------------O(n)

            for i in Dishes: -----------------------------------------O(n)

                if i[0]==j: ------------------------------------------O(1)

                    print('How much quantity do you want for', j) ----O(1)

                    quantity\_of\_j = int(input())----------------------O(1)

                    price = quantity\_of\_j \* i[1] ---------------------O(log(n))

                    push(total, price) -------------------------------O(1)

        print('Do you want to select some other cuisine?') -----------O(1)

b = input()---------------------------------------------------O(1)

        if b =='Yes' or b == 'yes': ----------------------------------O(1)

            Order\_processing(Menu)

        else: --------------------------------------------------------O(1)

            if b=='No' or b=='no': -----------------------------------O(1)

                print('This is your order:', lst) --------------------O(1)

                print('Thankyou for your order, you will recieve your order soon!') -------------------------------------------------------------------O(1)

        break

# (Order\_processing(Menu))

Total time complexity = n+n^2+nlogn

As, the most fastest growing term amoung them is O(n) so the total time complexity would be O(n)

**CODE FOR BILLING:**

# # Billing

def Counting\_price(total):

    count=0---------------------------------------------------------O(1)

    for i in total:-------------------------------------------------O(n)

        count+=i----------------------------------------------------O(1)

    Total=count-----------------------------------------------------O(1)

    print('Hello, your total bill is: ',Total)----------------------O(1)

    while True:-----------------------------------------------------O(n)

        Total\_money= int(input('Please pay the bill:'))-------------O(1)

        if Total\_money == (Total):----------------------------------O(1)

            print('Thank you for visiting!\n Hope to see you again:)')—0(1)

            break

        elif Total\_money > Total:-----------------------------------O(1)

            change = Total\_money - (Total)--------------------------O(1)

            print('Thank you for visiting, here is the change', change,'\nHope to see you again:)')--------------------------------------------------O(1)

            break

        else:-------------------------------------------------------O(1)

            c=(Total)-Total\_money-----------------------------------O(1)

            print('Sorry, but you need to pay ',c,'more\nThank you for visiting\n Hope to see you again:)')------------------------------------------O(1)

            break

# Counting\_price(total)

As the first for loop runs for n times and the second also runs for n times. So, n+n=2n, and as per the rule of time complexity we neglect the coefficient of the fastest growing term. Therefore, the total time complexity would be O(n).

**CODE FOR STORING DATA:**

#Saving data

data=name+':'+str(lst)+'\n' -----------------------------------------------O(1)

def saving\_data(data):

    with open("Customers\_Record.txt", "r+") as f:----------------------O(1)

        text=(f.read())------------------------------------------------O(n)

        f.write(data)--------------------------------------------------O(n)

# saving\_data(data)

Total time complexity would be O(n).

**CODE FOR ADDING AND DELETING THE DISHES:**

#Adding dishes

def add\_dishes(Menu, key, value):

    # We are making this function to help the admin to add dishes as per their convenience.

    Keys=list(Menu.keys())--------------------------------------O(1)

    for i in Keys:----------------------------------------------O(n)

        if i==key:----------------------------------------------O(1)

            push(Menu[key], value)------------------------------O(1)

    return Menu-------------------------------------------------O(1)

# print(add\_dishes(Menu, 'Beverages', ('Pomegranate Juice', 250)))

# Removing dishes

def remove(lst, value):

    for i in lst:----------------------------------------------O(n)

        if i[0] == value:--------------------------------------O(1)

            lst.remove(i)--------------------------------------O(1)

    return(lst)------------------------------------------------O(1)

##Deleting dishes

def delete\_dishes(Menu, key, dish\_to\_delete):

    # We are making this function to help the admin to delete dishes as per their convenience.

    Keys=list(Menu.keys())-------------------------------------O(1)

    for i in Keys:---------------------------------------------O(n)

        if i==key:---------------------------------------------O(1)

            (remove(Menu[key], dish\_to\_delete)) #using the remove code.---O(n)

            print(Menu)----------------------------------------O(1)

# print(delete\_dishes(Menu, 'Desi Dishes', 'Bihari Boti'))

**Remove function Time Complexity:**Total Time complexity for add\_dishes function would be O(n).

**Delete\_dishes** **Time Complexity:**As the outer loops run for n times and the inner fuction call runs for n times as well. So, the total Time complexity for delete dishes function would be O(n^2).

**CODE FOR RECOMMENDATION OF THE MOST ORDERED DISH:**

# Recommendation for the most ordered dish

def orders(filename):

    d={}------------------------------------------------------O(1)

    with open("Customers\_Record.txt") as f:--------------------O(n)

        for line in f:-------------------------------------O(n)

            key, val = line.split(":")-------------------O(1)

            s = val.replace("\n","")---------------------O(1)

            li=list(s[1:-1].split(','))------------------O(1)

            d[key]=li-------------------------------------O(1)

    return (d)------------------------------------------O(1)

print(orders('Customers\_Record.txt'))

d=orders(('Customers\_Record.txt'))----------------------O(n^2)

freq=dict()-----------------------------------------------O(1)

for i in d.keys():-----------------------------O(n)

    for j in d[i]:------------------------O(n)

        if j not in freq:----------------------O(1)

            freq[j]=1-------------------------O(1)

        else:

            freq[j]+=1-----------------------------O(1)

# print(freq)

values=list(freq.values())----------------------------O(1)

def recommendation(values):

    a=max(values)-------------------------------O(1)

    kv={}---------------------------------O(1)

    for k,v in freq.items():---------------O(n)

        if a==v:--------------------------O(1)

            kv[k]=v----------------O(1)

    print(kv)------------------O(1)

    print('This is the most ordered dish of our restaurant, would uou like to have it?')

    inp=input()

    if inp=='Yes' or inp=='yes':

        print('Please add this in your dishes')

    else:

        print('Okay:), Please provide your order!')

(recommendation(values))

Total time complexity would be O(n^2)

**Recommendation for the previous customers:**

##Recommending previous customers their previous dishes.

d=orders(('Customers\_Record.txt'))

def recommend\_previous\_dishes(d):

    for key in d:-------------------------------------O(n)

        if name == key:

            print(d[key])

            print('These are the dishes you ordered last time.\n We recommend you to order them again:)')

(recommend\_previous\_dishes(d))

Total time complexity will O(n)

**REVIEW FORM:**

def push(lst, item):

       lst.append(item)

       return lst

##REVIEW FORM:

print('Hello, please let us know how your experience about our restaurant by filling out this survey!')

print('please tell us your name:)')

name=input()

print('Rate the food quality between 1 to 5, 1 as worst and 5 as excellent')

rate=int(input())

if rate==1:

       print('worst')

elif rate==2:

       print('Okay, Okay')

elif rate==3:

       print('Average')

elif rate==4:

       print('Good')

else:

       print('Excellent')

a=str(rate)+' '+'to our food quality'+'\n'

print('Rate the taste of our food between 1 to 5, 1 as worst and 5 as excellent')

rate=int(input())

if rate==1:

       print('worst')

elif rate==2:

       print('Okay, Okay')

elif rate==3:

       print('Average')

elif rate==4:

       print('Good')

else:

       print('Excellent')

b=str(rate)+' '+'to the taste of our food'+'\n'

print('Rate the presentation of our food between 1 to 5, 1 as worst and 5 as excellent')

rate=int(input())

if rate==1:

       print('worst')

elif rate==2:

       print('Okay, Okay')

elif rate==3:

       print('Average')

elif rate==4:

       print('Good')

else:

       print('Excellent')

c= str(rate)+' '+'to the presentation of our food'+'\n'

print('Rate our Service between 1 to 5, 1 as worst and 5 as excellent')

rate=int(input())

if rate==1:

       print('worst')

elif rate==2:

       print('Okay, Okay')

elif rate==3:

       print('Average')

elif rate==4:

       print('Good')

else:

       print('Excellent')

d=str(rate)+' '+'to our Service'+'\n'

lst=[]

print('Do you want to give any other comments(Yes, No)?')

inp=input()

if inp=='No' or inp=='no':

       print('Okay!\nThankyou!')

       e='No any other comments'

       push(lst,e)

elif inp=='Yes' or inp=='yes':

       print('Type here')

       z=input()

       print(z)

       zz='The comment given:'+z

       push(lst, zz)

       print('Thankyou!')

lst=''.join(lst)

data='\n'+name+':'+'\n'+a+b+c+d+str(lst)+'\n'-----------------------O(1)

# print(data)

def saving\_data(data):

       with open("review file.txt", "r+") as f:---------------------O(1)

              text=(f.read())---------------------------------------O(n)

              f.write(data)-----------------------------------------O(n)

saving\_data(data)

Total complexity for the saving data function in the review form will be:

O(n)+O(n)=O(2n)

Coefficiant will be neglected.

O(n) .

**GOALS COMPLETED AND OUTCOMES:**

* Restaurant Menu was created in which cuisines represent keys of the dictionary and dishes and prices are the values in the form of list of tuples.
* Admin panel can add and delete food items from the Menu
* Billing
* File that stores the data (name, order, and cuisines) of every customer
* Review form
* File that stores reviews of customers
* Recommendation of most ordered dish and cuisine

**Challenges and outcome:**

* The part of my project that we found most challenging was Saving data in a file in which we were saving the information of the costumer which includes the name of the costumer along with the dishes ordered. This was challenging because we have’nt learned how to save data in a file in the class and for this we also asked the TA about it which helped us to understand its functioning.
* The other challenge that we faced on was reading the file data because we had to arrange the code in the form of dictionary to make the file read and to put the file data in the manageable form so that we can ask the customer of the menu in a proper form.