

18CS601

Foundations of Computer Science : Data Structures & Algorithms

CASE STUDY PROBLEMS # 1,2,5,8

DEEP PANCHOLI [CB.EN.P2CSE20011]

Problem #1

Objective Statement

Write a program Compute that computes the t-statistic to test whether two paired samples (x_1, x_2, \dots, x_N) and (y_1, y_2, \dots, y_N) have the same mean.

- The sample means μ_1 and μ_2 of the two samples are defined as:

$$\mu_1 = \frac{x_1 + x_2 + \dots + x_N}{N}, \quad \mu_2 = \frac{y_1 + y_2 + \dots + y_N}{N}$$

- The sample variances σ_1^2 and σ_2^2 of the two samples are defined as:

- The t-statistic of the paired samples is defined as:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 / N + \sigma_2^2 / N}}$$

$$\sigma_1^2 = \frac{(x_1 - \mu_1)^2 + (x_2 - \mu_1)^2 + \dots + (x_N - \mu_1)^2}{N - 1}$$
$$\sigma_2^2 = \frac{(y_1 - \mu_2)^2 + (y_2 - \mu_2)^2 + \dots + (y_N - \mu_2)^2}{N - 1}$$

A t-statistic value far from zero indicates greater statistical significance that the samples have different means.

Input and output specifications: The input from standard input consists of an integer N followed by N pairs of real numbers (the paired samples). The output to standard output is a single real number—the t-statistic of the paired samples.

Example

Example. For example, suppose that $N = 4$ and the paired samples are:

| i | x_i | y_i |
|---|-------|-------|
| 1 | 10.0 | 14.0 |
| 2 | 12.0 | 14.0 |
| 3 | 9.0 | 14.0 |
| 4 | 9.0 | 18.0 |

$$\mu_1 = \frac{10 + 12 + 9 + 9}{4} = 10, \quad \mu_2 = \frac{14 + 14 + 14 + 18}{4} = 15$$

$$\sigma_1^2 = \frac{(10 - 10)^2 + (12 - 10)^2 + (9 - 10)^2 + (9 - 10)^2}{3} = 2$$

$$\sigma_2^2 = \frac{(14 - 15)^2 + (14 - 15)^2 + (14 - 15)^2 + (18 - 15)^2}{3} = 4$$

$$t = \frac{10.0 - 15.0}{\sqrt{2/4 + 4/4}} = -4.08248290463863$$

Then, You may assume that the array arguments to `mean()`, `variance()`, and `tstat()` contain at least two elements (and that the two array arguments to `tstat()` have the same size).

Strategy used to solve the problem

1. Define a class Tstat.
2. Data members: two lists, l1 and l2.
3. Member functions:
 - get_mean : take a list of numbers as parameter and returns its arithmetic mean.
 - get_variance : takes a list of numbers as parameter and returns the variance of numbers in it.
 - get_tstat : takes 5 parameters: mean, variance of 2 lists and length of lists and returns the T-statistic value as per the formula.
4. Driver Code: takes the length of lists(n), and n pairs of numbers, creates two lists – one with the first values of pairs and the second with the second number of pairs and computes mean, variance and tstat using Tstat class.

Which Data Structure used and Why?

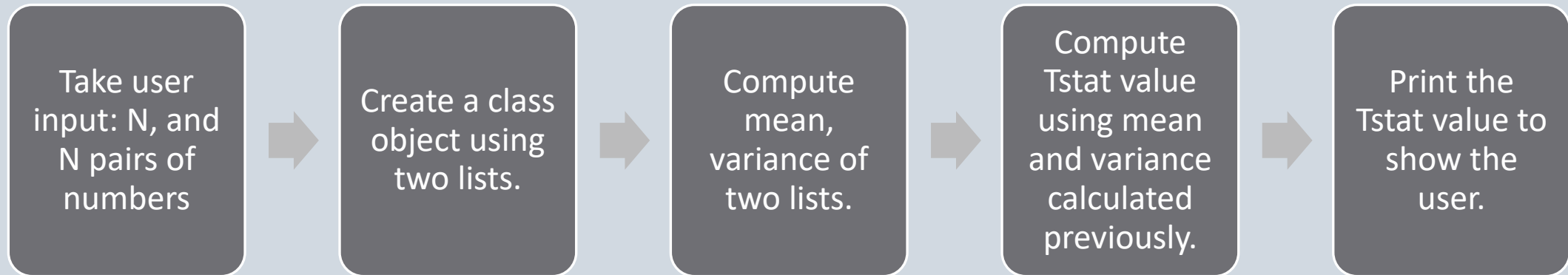
List:

The pairs of numbers input by user are stored as lists.

Why?

Because python provides various inbuilt packages to perform calculations on lists.

Algorithm



Expected Outcome

The program is expected to show the correctly calculated value of T-statistic value of two lists input by user.

->Code Demo and Output

Problem #2

Objective Statement

Your program must be organized as a library of methods with the following

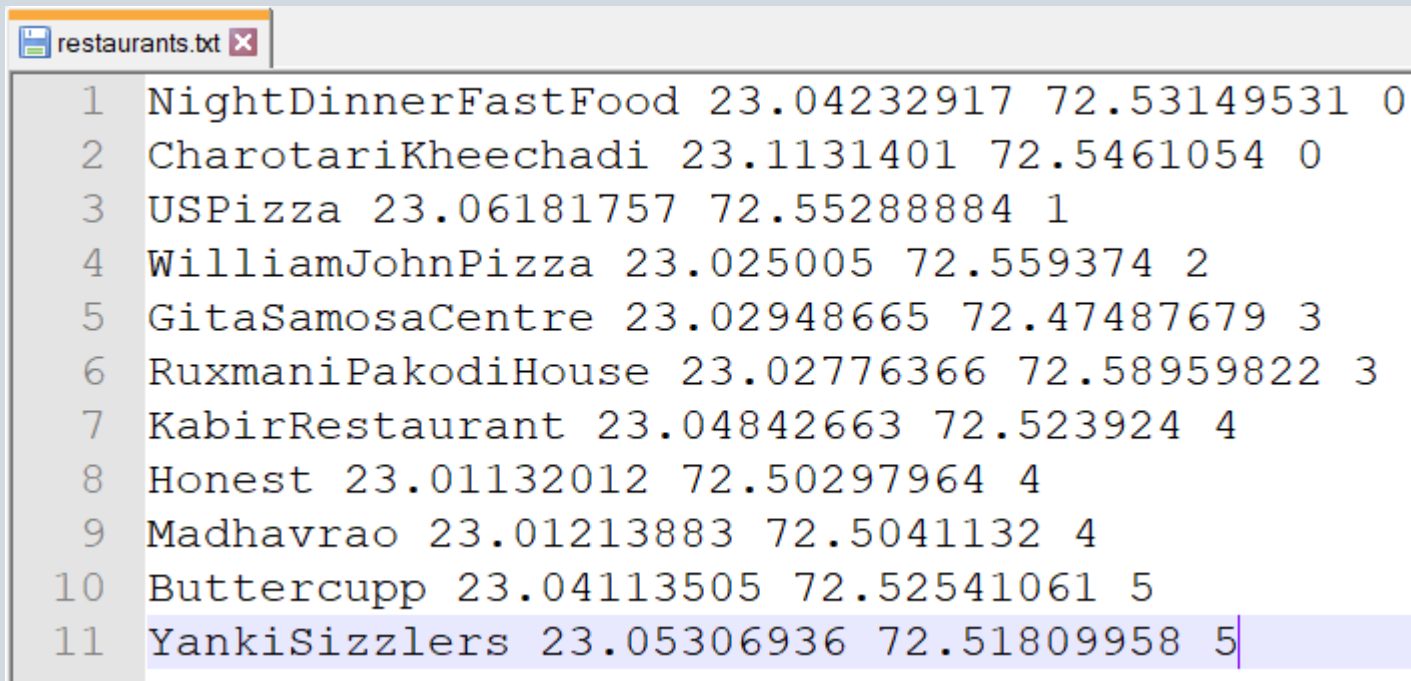
API:

class FindFood

| | |
|---|---|
| double dist(double x1, double y1, double x2, double y2) | // Calculate the spherical distance between (x1, y1) and (x2, y2). // Compute the distance using the Great Circle formula. |
| int findNext(double x, double y, double[] lat, double[] lon) | // Given my latitude and longitude as (x, y) and the latitudes and longitudes of the restaurants as (lat[i], lon[i]), // return the integer index i of the restaurant closest to me. |
| void markFound(int f, double[] lat, double[] lon) | // Given an index f that identifies a restaurant and the latitudes and longitudes of all the restaurants, change the latitude and longitude // of restaurant f to Double.POSITIVE_INFINITY so that f will be ignored by findNext() |
| void main(String[] args) | // 1. Input user position and minumum rating from the command- line. // 2. Read each restaurant's latitude, longitude, name, and rating from standard input and store in parallel arrays. // 3. Find the closest restaurant and, if it has an acceptable rating, print its name, distance and rating. Mark it as found, and repeat 3 until all acceptable restaurants are printed. |

Objective Statement

Data: The restaurant data file has the number of restaurants on the first line. Each subsequent line has the name, latitude, longitude, and rating for one restaurant. All ratings are between 1 and 5, inclusive. Restaurant names do not have spaces. For example:

A screenshot of a text editor window titled 'restaurants.txt'. The window displays a list of 11 restaurants, each on a new line. The first line is highlighted in light blue. The data for each restaurant is as follows:

| Line | Restaurant Name | Latitude | Longitude | Rating |
|------|---------------------|-------------|-------------|--------|
| 1 | NightDinnerFastFood | 23.04232917 | 72.53149531 | 0 |
| 2 | CharotariKheechadi | 23.1131401 | 72.5461054 | 0 |
| 3 | USPizza | 23.06181757 | 72.55288884 | 1 |
| 4 | WilliamJohnPizza | 23.025005 | 72.559374 | 2 |
| 5 | GitaSamosaCentre | 23.02948665 | 72.47487679 | 3 |
| 6 | RuxmaniPakodiHouse | 23.02776366 | 72.58959822 | 3 |
| 7 | KabirRestaurant | 23.04842663 | 72.523924 | 4 |
| 8 | Honest | 23.01132012 | 72.50297964 | 4 |
| 9 | Madhavrao | 23.01213883 | 72.5041132 | 4 |
| 10 | Buttercupp | 23.04113505 | 72.52541061 | 5 |
| 11 | YankiSizzlers | 23.05306936 | 72.51809958 | 5 |

Example

Suppose I'm at (40.75, -73.98), and I want to find restaurants near me with ratings of at least 3. Then, my program should behave as follows:

```
% java FindFood 40.75 -73.98 3 < 5restaurants.txt
```

TheRussianTeaRoom has a rating of 3

Distance: 1.292930 miles

TheCorneliaStreetCafe has a rating of 5

Distance: 1.787035 miles

GreenwichVillageBistro has a rating of 5

Distance: 1.793825 miles

Delmonico's has a rating of 4

Distance: 3.245044 miles

Strategy used to solve the problem

1. Define a class Restaurants.
2. Data members: two variables, latitude and longitude of user location.
3. Member functions:
 - dist : takes 2 pairs of latitude and longitude as parameter and returns the distance according to Great Circle Formula for Land.
 - findnext: takes two lists of numbers – one with latitudes and other with longitudes of restaurants as fetched from the text file as parameter and returns a list of distance with the user location.
 - markfound: takes 2 parameters: min_rating required by user and a list of ratings of all restaurants and returns the list of name of restaurants whose rating is above min_rating.
4. Driver Code: takes latitude and longitude from user creates an object of class Restaurants with it and computes distance of user from all the restaurants and finds min_rating restaurants with distance from the user.

Which Data Structure used and Why?

Pandas DataFrame: To store the data of restaurants read from text file.

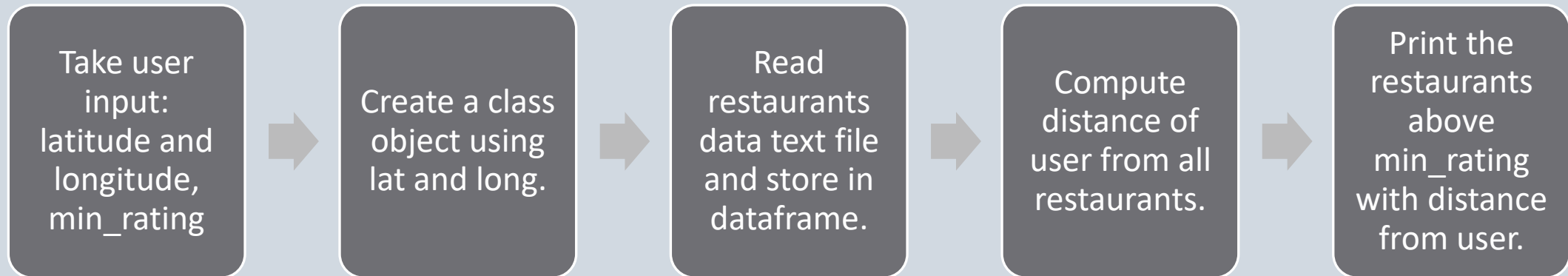
List: To store the index wise computed distances of user from all the restaurants.

Why?

Because python provides various inbuilt packages to perform calculations on lists.

Pandas Data Frames are easy to work with as individual columns can be processed and data can be easily fetched from it.

Algorithm



Expected Outcome

The program is expected to process the correctly calculated distance of user from all the restaurants and show the ones with rating \geq min_rating.

->Code Demo and Output

Problem #5

Objective Statement

HVAC Load:

In this project you will calculate the heating and cooling loads for a typical residence. The wintertime heating load is all of the heat that flows out of the building when the outside temperature is nearly its coldest and when there is no free heat provided by sunshine or internal lights or appliances. The coldest time is typically in the middle of January. The summertime cooling load is all of the heat that flows into the building when the outside temperature is nearly its hottest, the outside humidity is high, there are no clouds to block the sun, and the building is occupied with normal lighting and appliance usage. The hottest time is usually about 3:00 PM in the middle of July.

Example

HVAC Load Calculation:

```
Enter total roof area in square ft: 1500
Enter building perimeter in ft: 140
Enter building height in ft: 18
total window area in square ft: 400
Enter west window area in square ft: 80
Enter roof U-value: 0.04
Enter wall U-value: 0.10
Enter window U-value: 0.5
Enter winter outdoor temperature in deg F: 2.0
Enter summer outdoor temperature in deg F: 100
Enter summer outdoor humidity ratio: 0.013
Enter infiltration in CFM: 200
Enter number of people present: 3
Enter electrical use in Watts: 1500
Heating Load = 62608.0 BTU per hour.
Cooling Load = 45354.100000000006 BTU per hour.
```

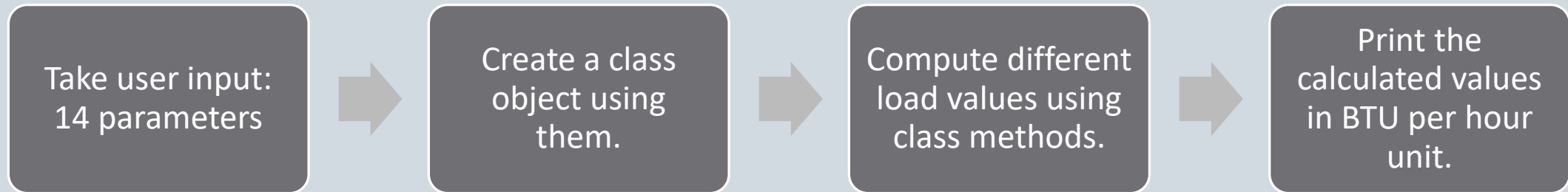
Strategy used to solve the problem

1. Define a class HVACLoad.
2. Data members: 14 variables, taken as user input.
3. Member functions:
 - findHeatingLoad: calculates heating load based on formulas provided and user inputs.
 - findCoolingLoad: calculates latent cooling load
 - findCoolingSensibleLoad: calculates sensible cooling load value as per the formula.
4. Driver Code: takes 14 parameters as input from user and calculates load values using above methods in class.

Which Data Structure used and Why?

No Specific Data structure is used here because class object instantiated with user input values does the necessary work.

Algorithm



Expected Outcome

The program is expected to show the correctly calculated values of Heating load, cooling load and sensible cooling load from the values input by user.

->Code Demo and Output

Problem #8

Objective Statement

FollowMe is a popular video game:

- The computer display a sequence of values, and then asks the player to reproduce the same sequence.
- Points are scored if the entire sequence was produced in order.

Example

```
Level 1
Enter the list elements in order as you remember:
39
you scored 1/1 points in this level.
Level 2
Enter the list elements in order as you remember:
73
26
you scored 2/2 points in this level.
Level 3
Enter the list elements in order as you remember:
46
15
20
you scored 3/3 points in this level.
```

```
Level 4
Enter the list elements in order as you remember:
96
70
88
99
you scored 4/4 points in this level.
Level 5
Enter the list elements in order as you remember:
90
53
48
52
42
you scored 4/5 points in this level.
Your cumulative Score is: 14
```

Strategy used to solve the problem

1. Define a class Queue to be implemented.
2. Data members: one list – to be used as queue
3. Member functions:
 - enqueue: take a number and enqueue in the front of queue.
 - dequeue : dequeues and element from the front of queue.
 - first: returns the first element in the queue without dequeue operation.

Which Data Structure used and Why?

Queue:

The numbers list generated randomly is stored in a queue.

Why?

Because we want to check if the user input matches the random list in sequence.

So when we check element by element, if it matches, increment score by 1 and dequeue the element from queue.

Algorithm



Expected Outcome

The program is expected to show the correctly calculated score of user playing the game.

->Code Demo and Output

Thank You!