COMP SCI 4094/4194/7094 - Distributed Databases and Data Mining
Assignment 3

DUE: 23:59 Thursday 28th October

Important Notes

- Handins:
  - The deadline for submission of your assignment is **23:59 Thursday 28th October , 2021**.
  - You must do this assignment individually and make individual submissions.
  - Your program should be coded in **C++** and pass test runs on 4 test files. The sample input and output files are downloadable in "Assignments" of the course home page (`https://myuni.adelaide.edu.au/courses/64886/assignments/238277`).
  - You need to use `svn` to upload and run your source code in the web submission system following "Web-submission instructions" stated at the end of this sheet. You should attach your name and student number in your submission.
  - Late submissions will attract a penalty: the maximum mark you can obtain will be reduced by 25% per day (or part thereof) past the due date or any extension you are granted.
- Marking scheme:
  - 16 marks for testing on 4 random tests: 4 marks per test.
    For **undergraduate students**, We want your code cluster the Flows by **Manhattan distance**:
    1 mark for Flow.txt
    3 marks for KMedoids.txt (3 marks for absolute value)

    For **postgraduate students**, you should design a suitable code structure or API to make this code expect more flexible. We want your code can easily change from **Manhattan distance** to **Euclidean distance**. You should write this functions on you code:
    1 mark for Flow.txt
    2 marks for KMedoids.txt (Use Manhattan distance to cluster)
    1 mark for KMedoidsE.txt (Use Euclidean distance to cluster)
  - 4 marks for the code styles. (Put your id, name, **postgraduate or undergraduate** on the code header comment)
  - **Note:** If it is found your code did not implement the required computation tasks in this assignment, you will receive zero mark regardless of the correctness of testing output.

If you have any questions, please send them to the student discussion forum. This way you can all help each other and everyone gets to see the answers.

The assignment

In this assignment you are required to code a traffic packet clustering engine to cluster the raw network packet to different applications, such as http, smtp. To accomplish this assignment, **a data preprocessing module** and **a clustering module** should be implemented.

You will have two input files, and you should print two(undergraduate) or three(postgraduate) output files.

0.1 Input File:

The input file1 contains a distance threshold and the raw network packet information, that is, seven attributes of a packet: source address, source port, destination address, destination port, protocol, arrival time, and packet length.

1. Input file1.txt is sample traffic flow information, which looks like:

src addr	src port	dst addr	dst port	protocol	arrival time	packet length
202.234.224.254	49880	31.65.181.210	80	6	115258	52
202.234.224.254	49880	31.65.181.210	80	6	115307	52
202.234.35.144	55256	74.39.124.220	443	6	115310	46
119.188.179.82	50592	150.79.7.129	80	6	115314	40
202.234.224.254	49880	31.65.181.210	80	6	115341	52
119.188.179.82	50592	150.79.7.129	80	6	115350	40
119.188.179.82	50592	150.79.7.129	80	6	115363	40

2. Input file2.txt has a number K, and on the next line include K integer numbers represent an initial set of K medoids, which looks like:

   1 (k=1)
   0 (Start from index 0, as the initial start medoid)

0.2 Output File:

You should print out:

for **undergraduate students**:

1. Flow.txt (for data preprocessing result, 1 mark per test)
2. KMedoids.txt (for clustering result by Manhattan distance, 2 marks for absolute value, 1 mark for details).

for **undergraduate students**:

1. Flow.txt (for data preprocessing result, 1 mark per test)
2. KMedoids.txt (for clustering result by Manhattan distance, 2 marks).
3. KMedoidsE.txt (for clustering result by Euclidean distance, 1 mark).

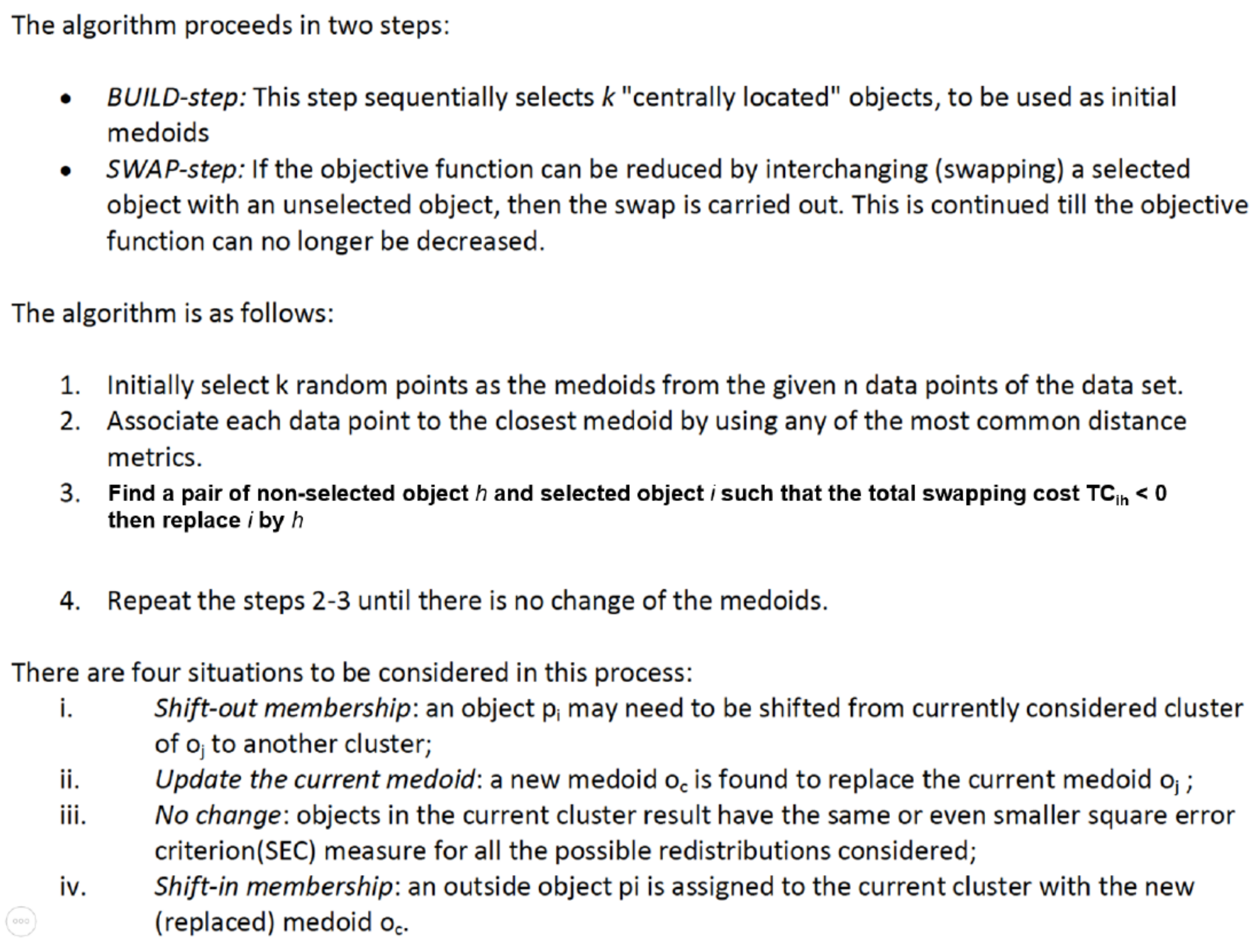
What you need to do:

In the **data preprocessing module**, your program should prepare the flow data for clustering by the raw packet data, two steps are involved: you need to firstly merge the packets into flows by the rule: a network flow includes at least **TWO** packets with same source address, source port, destination address, destination port, and protocol, then calculate two clustering features: average transferring time and the average packet length of a flow.

In the **clustering module**, you need to apply k-medoids algorithm (course slides Chapter 10, not the book's random method) to find the minimum number of clusters that the sum of the distance of each flow to its centroid is less than the given threshold. Note: the clustering features come from data preprocessing module, the distance measurement is Mannhaton distance.

For your convenience, below is the framework of the k-medoids algorithm which you should follow:

We will use PAM algorithm on ClusBasic.pdf page 20: `https://myuni.adelaide.edu.au/courses/64886/discussion_topics/602515`



The algorithm proceeds in two steps:

- *BUILD-step:* This step sequentially selects *k* "centrally located" objects, to be used as initial medoids
- *SWAP-step:* If the objective function can be reduced by interchanging (swapping) a selected object with an unselected object, then the swap is carried out. This is continued till the objective function can no longer be decreased.

The algorithm is as follows:

1. Initially select k random points as the medoids from the given n data points of the data set.
2. Associate each data point to the closest medoid by using any of the most common distance metrics.
3. **Find a pair of non-selected object *h* and selected object *i* such that the total swapping cost $TC_{ih} < 0$ then replace *i* by *h***
4. Repeat the steps 2-3 until there is no change of the medoids.

There are four situations to be considered in this process:

i. *Shift-out membership*: an object p_i may need to be shifted from currently considered cluster of o_j to another cluster;

ii. *Update the current medoid*: a new medoid o_c is found to replace the current medoid o_j ;

iii. *No change*: objects in the current cluster result have the same or even smaller square error criterion(SEC) measure for all the possible redistributions considered;

iv. *Shift-in membership*: an outside object pi is assigned to the current cluster with the new (replaced) medoid o_c.

Example

Sample traffic flow information

src addr	src port	dst addr	dst port	protocol	arrival time	packet length
202.234.224.254	49880	31.65.181.210	80	6	115258	52
202.234.224.254	49880	31.65.181.210	80	6	115307	52
202.234.35.144	55256	74.39.124.220	443	6	115310	46
119.188.179.82	50592	150.79.7.129	80	6	115314	40
202.234.224.254	49880	31.65.181.210	80	6	115341	52
119.188.179.82	50592	150.79.7.129	80	6	115350	40
119.188.179.82	50592	150.79.7.129	80	6	115363	40

Data preprocessing module

Firstly, we should identify different flows (different flows have different source and destination addresses).
In the above traffic flow information, there are two flows: The first, second, and fifth packet belong to the first flow(index is 0); the fourth, sixth, and seventh packet belong to the second flow(index is 1).

The Average transferring time of first flow = ((the arrival time of fifth packet - the arrival time of second packet) + (the arrival time of second packet - the arrival time of first packet)) $\div$ (3 - 1) = ((115341 - 115307) + (115307 - 115258)) $\div$ 2 = 41.5. The Average length of first flow = ($\sum$ packet length) $\div$ 3 = (52 + 52 + 52) $\div$ 3 = 52. Similarly, the Average transferring time of second flow = 24.5, the average length of second flow = 40.
(arrival time is microsecond(μs))

Clustering module

We use Manhattan distance to measure the distance between flows. In our sample, the distance between the two flows is $|41.5 - 24.5| + |52 - 40|$.

Example Output

At begin you should output the flow after Data preprocessing module, include index, average transferring time x value and average length y value.
ID X Y
In this case, **Flow.txt** should print:
0 41.50 52.00
1 24.50 40.00

Rounding numbers (X,Y) to 2 decimal place. You can use:

$$cout << fixed << setprecision(2) << 3.1415926;$$

or

$$printf("\%0.2f", 3.1415926);$$

After doing KMedoid, you will get K clusters.

You should provide KMedoids.txt file:

It includes K+2 lines. First line is absolute-error criterion (First line it important, other lines is help you to debug.). Next one line include K medoids' index. Following each line have several flow index (In order of number) represent each medoid includes which flows.
29 (Absolute-error of the cluster, 2 decimal places)

0 (Medoid is 0)
0 1 (This cluster include 2 flows index 0 and index 1)

For **postgraduate students**, you should design a suitable code structure or API. This code is expected more flexible. It should be easily changed from Manhattan distance to **Euclidean distance**. You should write this functions on you code.
Tips: you can use object-oriented, class-based, or other well-organized methods.

You should print KMedoidsE.txt, the structure is same as KMedoids.txt `https://en.wikipedia.org/wiki/Euclidean_distance`

Web-submission instructions

- First, type the following command, all on one line (replacing xxxxxxx with your student ID):
  svn mkdir --parents -m "DDDM"
  https://version-control.adelaide.edu.au/svn/axxxxxxx/2021/s2/dddm/assignment3

- Then, check out this directory and add your files:
  svn co https://version-control.adelaide.edu.au/svn/axxxxxxx/2021/s2/dddm/assignment3
  cd assignment3
  svn add KMedoidsUG.cpp (or KMedoidsPG.cpp)
  ...
  svn commit -m "assignment3 solution"

- Next, go to the web submission system at:
  `https://cs.adelaide.edu.au/services/websubmission/`
  Navigate to 2021, Semester 2, Distributed Databases and Data Mining, Assignment 3. Then, click Tab "Make Submission" for this assignment and indicate that you agree to the declaration. The automark script will then check whether your code compiles. You can make as many resubmissions as you like. If your final solution does not compile you won't get any marks for this solution.

- **Note:**

  1. Please follow the forms in sample output files.
  2. Your local file path will not work with our web-submission system.
  3. We prepared ten test files in web-submission system, when you submit your program, random test files will be allocated for you.
  4. The auto-marker script compiles and runs named "KMedoidsUG.cpp" or "KMedoidsPG.cpp" by using following command(please only submit **one** cpp file, name KMedoidsUG.cpp or KMedoidsPG.cpp):
     g++ -std=c++11 KMedoidsUG.cpp -o runKMedoids (for undergraduate students)
     g++ -std=c++11 KMedoidsPG.cpp -o runKMedoids (for postgraduate students)
     ./runKMedoids network_packets.txt initial_medoids.txt
     In this assignment, you need to read two files network_packets.txt (network packets traffic information) and initial_medoids.txt (initial medoids) which are generated randomly by the system.
  5. Absolute-error is the total manhattan distances. K-medoid is aiming to narrow down the distance between the each point and their clusters.
  6. Your code should follow default order of the K-Medoid algorithm. If you not use the default order. It may cause your absolute vaule is right but KMedoidsdeails.txt is wrong.
  7. If the answer is around the standard absolute value, we will accept this answers. Eg: standard absolute value is 8223.23 and your absolute value is 8222.11, we will accept your answer.
  8. IF you have any questions on assignment 3 you can ask in this link: `https://myuni.adelaide.edu.au/courses/64886/discussion_topics/602515` . Tips: If you have accuracy problem in final absolute-error, fristly, you can try to resubmit code(because data is random generated). If that not fix the accuracy problem, you can put it on discussion board, I will manual judge it.

You should print two or three output files as shown in the following two examples.

Example1

input:File1.txt

src addr	src port	dst addr	dst port	protocol	arrival time	packet length
202.234.224.254	49880	31.65.181.210	80	6	115258	52
202.234.224.254	49880	31.65.181.210	80	6	115307	52
202.234.35.144	55256	74.39.124.220	443	6	115310	46
119.188.179.82	50592	150.79.7.129	80	6	115314	40
202.234.224.254	49880	31.65.181.210	80	6	115341	52
119.188.179.82	50592	150.79.7.129	80	6	115350	40
119.188.179.82	50592	150.79.7.129	80	6	115363	40

input:File2.txt

```
1
0
```

output:Flow.txt

```
0 41.50 52.00
1 24.50 40.00
```

output:KMedoids.txt

```
29.00
0
0 1
```

output for postgraduate:KMedoidsE.txt

```
20.81
0
0 1
```

Example2

input:file1.txt

```
src addr  src port dst addr  dst port  protocol  arrival time    packet length
61.43.24.146 80 133.227.178.71 55651 6 115164 1500
223.139.34.184 57258 203.146.250.47 80 6 115167 40
118.162.252.133 8100 150.79.7.129 80 6 115178 52
163.39.157.71 52864 199.252.216.15 443 6 115181 436
125.96.202.102 80 202.31.174.9 36122 6 115185 185
202.234.224.254 49880 31.65.181.210 80 6 115189 52
61.211.145.45 61611 150.79.7.129 80 6 115222 40
202.234.224.254 49880 31.65.181.210 80 6 115226 52
163.39.157.71 52864 199.252.216.15 443 6 115230 1426
163.39.157.71 52865 199.252.216.15 443 6 115233 436
118.91.103.40 53186 150.79.7.129 80 6 115244 52
133.244.153.246 54194 165.143.250.152 443 6 115247 52
202.234.224.254 49880 31.65.181.210 80 6 115251 52
163.39.157.71 52865 199.252.216.15 443 6 115254 1426
202.234.224.254 49880 31.65.181.210 80 6 115258 52
202.234.224.254 49880 31.65.181.210 80 6 115307 52
202.234.35.144 55256 74.39.124.220 443 6 115310 378
119.188.179.82 50592 150.79.7.129 80 6 115314 40
202.234.224.254 49880 31.65.181.210 80 6 115320 52
202.234.224.254 49880 31.65.181.210 80 6 115326 52
202.234.224.254 49880 31.65.181.210 80 6 115331 52
202.234.35.144 50070 173.199.56.254 80 6 115335 40
54.221.15.83 443 150.79.179.172 60804 6 115349 52
202.145.203.99 443 163.39.7.122 53326 6 115435 1500
133.227.171.14 52147 121.131.234.16 80 6 115439 818
131.14.216.241 24153 54.43.88.212 80 6 115443 1496
202.145.203.99 443 163.39.7.122 53326 6 115447 1188
203.146.250.47 80 5.98.62.124 47610 6 115461 1460
69.192.0.189 80 202.234.225.187 59368 6 115469 1500
69.192.0.189 80 202.234.225.187 59368 6 115491 1500
202.234.228.45 58507 38.249.43.123 443 6 115494 1500
163.39.110.212 49700 204.93.161.172 443 6 115501 819
126.71.29.111 61782 203.146.247.176 80 6 115512 40
126.71.29.111 61782 203.146.247.176 80 6 115516 40
131.14.216.241 24153 54.43.88.212 80 6 115519 1496
203.146.250.47 80 113.63.133.249 39564 6 115573 1500
202.231.242.67 49448 131.226.8.6 80 6 115576 249
157.210.227.245 60827 66.36.161.252 80 6 115580 52
203.146.250.47 80 113.63.133.249 39564 6 115584 1500
69.192.0.189 80 202.234.225.187 59368 6 115588 1500
69.192.0.189 80 202.234.225.187 59368 6 115597 1500
175.84.22.21 41639 150.42.176.170 54756 6 115601 60
219.80.177.15 33814 150.79.7.129 80 6 115605 52
202.234.228.45 58507 38.249.43.123 443 6 115609 1500
131.14.216.241 24153 54.43.88.212 80 6 115664 1496
163.39.157.71 52867 199.252.216.15 443 6 115751 1426
163.39.157.71 52867 199.252.216.15 443 6 115755 436
```

```
163.39.157.71 52864 199.252.216.15 443 6 115763 436
133.244.153.246 54194 165.143.250.152 443 6 115766 52
163.39.157.71 52864 199.252.216.15 443 6 115809 1426
131.14.216.241 24153 54.43.88.212 80 6 115815 1496
202.234.35.13 52171 185.213.144.150 80 6 115831 52
173.199.56.233 80 202.234.224.241 59801 6 115878 1500
173.199.56.233 80 202.234.224.241 59801 6 115893 1500
113.113.137.159 61396 150.79.7.129 80 6 115904 40
199.252.216.15 443 163.39.157.71 52864 6 115907 64
131.14.216.241 24153 54.43.88.212 80 6 115991 1496
199.252.216.15 443 163.39.157.71 52864 6 116014 52
133.244.153.246 54194 165.143.250.152 443 6 116049 52
96.227.76.37 3242 133.250.150.37 445 6 116075 48
131.14.216.241 24153 54.43.88.212 80 6 116084 1496
96.16.24.215 443 202.234.35.13 62476 6 116222 60
163.39.157.71 52865 199.252.216.15 443 6 116226 1426
131.14.216.241 24153 54.43.88.212 80 6 116229 1496
163.39.157.71 52865 199.252.216.15 443 6 116275 436
131.14.216.241 24153 54.43.88.212 80 6 116279 495
182.158.75.63 80 133.244.234.48 50169 6 116287 1490
61.210.137.135 56413 150.79.7.129 63190 6 116291 40
163.39.157.71 52867 199.252.216.15 443 6 116298 436
163.39.157.71 52867 199.252.216.15 443 6 116329 1426
133.244.153.246 54194 165.143.250.152 443 6 116333 52
131.14.188.92 34705 204.93.161.172 443 6 116349 52
211.73.188.247 443 202.234.35.13 36955 6 116358 1500
211.73.188.247 443 202.234.35.13 36955 6 116365 1500
211.73.188.247 443 202.234.35.13 36955 6 116400 1500
211.73.188.247 443 202.234.35.13 36955 6 116404 1500
126.71.29.111 61782 203.146.247.176 80 6 116415 40
202.234.228.45 58507 38.249.43.123 443 6 116423 1500
203.146.250.47 80 199.48.187.153 58554 6 116427 52
133.244.153.246 56862 31.65.185.141 443 6 116484 40
23.225.11.237 80 202.31.174.9 18622 6 116498 1430
23.225.11.237 80 202.31.174.9 18622 6 116501 1430
173.199.56.233 80 202.234.224.241 59801 6 116518 1500
173.199.56.233 80 202.234.224.241 59801 6 116522 1500
182.104.251.244 64598 150.79.7.129 80 6 116525 40
182.104.251.244 64598 150.79.7.129 80 6 116528 40
133.244.153.246 56862 31.65.185.141 443 6 116539 40
173.199.56.233 80 202.234.224.241 59801 6 116542 1500
173.199.56.233 80 202.234.224.241 59801 6 116566 1500
182.104.251.244 64598 150.79.7.129 80 6 116569 40
202.234.228.45 58507 38.249.43.123 443 6 116573 1500
173.199.56.233 80 202.234.224.241 59801 6 116576 1500
133.244.153.246 54194 165.143.250.152 443 6 116582 52
173.199.56.233 80 202.234.224.241 59801 6 116586 1500
106.127.152.45 56799 150.79.7.129 80 6 116589 40
124.44.132.23 443 202.231.242.67 49557 6 116601 294
211.3.241.186 14457 150.79.7.129 80 6 116605 40
223.139.34.184 57258 203.146.250.47 80 6 116610 40
```

```
200.98.164.214 3966 133.250.168.37 445 6 116615 48
199.252.216.15 443 163.39.157.71 52864 6 116623 52
199.252.216.15 443 163.39.157.71 52864 6 116630 52
133.244.153.246 56862 31.65.185.141 443 6 116641 40
61.43.24.146 80 133.227.178.71 55651 6 116645 1500
61.43.24.146 80 133.227.178.71 55651 6 116651 1500
223.25.5.131 64680 150.79.7.129 80 6 116654 40
175.167.20.236 10595 150.79.176.180 54762 6 116658 52
107.133.162.38 443 163.39.5.198 57375 6 116672 569
183.172.222.56 39620 150.79.7.129 80 6 116675 52
202.234.228.45 58507 38.249.43.123 443 6 116678 1500
183.172.222.56 39620 150.79.7.129 80 6 116682 52
183.172.222.56 39620 150.79.7.129 80 6 116688 52
183.172.222.56 39620 150.79.7.129 80 6 116692 52
183.172.222.56 39620 150.79.7.129 80 6 116696 52
183.172.222.56 39620 150.79.7.129 80 6 116699 52
183.172.222.56 39620 150.79.7.129 80 6 116703 52
183.172.222.56 39620 150.79.7.129 80 6 116706 52
183.172.222.56 39620 150.79.7.129 80 6 116709 52
183.172.222.56 39620 150.79.7.129 80 6 116713 52
183.172.222.56 39620 150.79.7.129 80 6 116716 52
133.244.153.246 56862 31.65.185.141 443 6 116727 40
203.146.253.28 6881 60.26.1.79 45729 6 116741 52
27.178.159.198 4419 150.79.7.129 80 6 116748 40
183.172.222.56 39620 150.79.7.129 80 6 116751 52
183.172.222.56 39620 150.79.7.129 80 6 116755 52
183.172.222.56 39620 150.79.7.129 80 6 116759 52
183.172.222.56 39620 150.79.7.129 80 6 116762 52
163.39.157.71 52864 199.252.216.15 443 6 116766 1426
163.39.157.71 52864 199.252.216.15 443 6 116769 436
133.244.153.246 56862 31.65.185.141 443 6 116773 40
65.119.5.150 80 150.79.7.11 52758 6 116777 192
182.104.251.244 64598 150.79.7.129 80 6 116781 40
163.39.157.71 52865 199.252.216.15 443 6 116788 436
126.71.29.111 61782 203.146.247.176 80 6 116796 40
202.234.228.45 58507 38.249.43.123 443 6 116801 1500
133.244.153.246 56862 31.65.185.141 443 6 116804 40
163.39.157.71 52865 199.252.216.15 443 6 116811 1426
163.39.157.71 52867 199.252.216.15 443 6 116818 436
61.43.24.146 80 133.227.178.71 55651 6 116822 1500
61.43.24.146 80 133.227.178.71 55651 6 116831 1500
126.71.29.111 61782 203.146.247.176 80 6 116838 40
133.244.153.246 54194 165.143.250.152 443 6 116841 52
163.39.157.71 52867 199.252.216.15 443 6 116844 1426
133.244.153.246 56862 31.65.185.141 443 6 116851 40
199.252.216.15 443 163.39.157.71 52864 6 116860 64
199.252.216.15 443 163.39.157.71 52864 6 116863 52
61.43.24.136 80 133.227.178.71 55658 6 116871 1500
202.234.228.45 58507 38.249.43.123 443 6 116875 1500
203.146.250.47 80 199.48.187.153 58554 6 116878 990
61.43.24.136 80 133.227.178.71 55658 6 116882 1500
```

```
40.17.153.225 443 133.244.144.247 22150 6 116885 434
223.139.34.184 57258 203.146.250.47 80 6 116888 40
223.139.34.184 57258 203.146.250.47 80 6 116892 40
133.244.153.246 56862 31.65.185.141 443 6 116898 40
36.10.160.187 64334 150.79.177.11 62064 6 116915 40
182.158.75.33 80 157.210.199.11 11540 6 116918 64
175.161.50.49 61316 150.79.7.129 80 6 116921 52
133.244.153.246 56862 31.65.185.141 443 6 116925 40
202.234.228.45 58507 38.249.43.123 443 6 116935 1500
133.244.153.246 54194 165.143.250.152 443 6 117021 52
133.244.153.246 56862 31.65.185.141 443 6 117028 40
23.238.55.225 80 202.234.35.13 54750 6 117031 1500
23.238.55.225 80 202.234.35.13 54750 6 117034 1500
133.244.153.246 56862 31.65.185.141 443 6 117038 40
113.150.148.134 9051 150.42.177.43 54756 6 117048 52
133.244.153.246 56862 31.65.185.141 443 6 117111 40
113.5.21.232 5328 150.79.7.129 80 6 117125 40
163.39.157.71 52864 199.252.216.15 443 6 117129 1426
163.39.157.71 52864 199.252.216.15 443 6 117133 436
163.39.157.71 52865 199.252.216.15 443 6 117136 436
133.244.153.246 56862 31.65.185.141 443 6 117193 40
1.106.21.96 1946 150.79.7.129 80 6 117204 40
133.244.153.246 54194 165.143.250.152 443 6 117207 52
60.36.215.88 51464 150.79.7.129 80 6 117212 52
163.39.157.71 52865 199.252.216.15 443 6 117215 1426
173.199.56.233 80 202.234.224.241 59801 6 117218 1500
173.199.56.233 80 202.234.224.241 59801 6 117225 1500
133.244.153.246 56862 31.65.185.141 443 6 117245 40
202.234.227.137 58409 89.57.134.9 80 6 117248 40
69.192.0.189 80 202.234.225.187 59368 6 117251 1500
202.234.227.137 58409 89.57.134.9 80 6 117258 40
69.192.0.189 80 202.234.225.187 59368 6 117261 1500
202.234.227.137 58076 89.57.134.158 80 6 117266 40
131.14.158.108 62531 216.19.170.177 80 6 117269 40
202.234.227.137 58409 89.57.134.9 80 6 117301 40
23.234.243.99 443 203.146.254.83 61708 6 117304 52
133.244.153.246 56862 31.65.185.141 443 6 117311 40
199.252.216.15 443 163.39.157.71 52864 6 117316 64
199.252.216.15 443 163.39.157.71 52864 6 117319 52
23.234.243.101 80 203.146.254.83 61718 6 117324 52
211.3.241.186 14457 150.79.7.129 80 6 117331 40
133.227.127.204 53917 103.238.115.79 80 6 117335 40
61.111.37.246 49473 150.79.7.129 80 6 117357 52
133.244.153.246 56862 31.65.185.141 443 6 117371 40
23.234.243.101 80 203.146.254.83 61712 6 117375 52
23.234.243.99 443 203.146.254.83 61711 6 117381 52
131.14.92.245 60000 54.20.141.183 443 6 117384 40
101.105.131.251 62325 203.146.240.134 80 6 117388 52
118.91.103.40 53186 150.79.7.129 80 6 117467 52
118.91.103.40 53186 150.79.7.129 80 6 117471 52
23.234.243.101 80 203.146.254.83 61719 6 117474 52
```

```
23.234.243.99 443 203.146.254.83 61710 6 117478 52
133.244.153.246 56862 31.65.185.141 443 6 117481 40
31.65.185.129 443 202.31.174.9 51205 6 117484 1430
31.65.185.129 443 202.31.174.9 51205 6 117487 1430
31.65.185.129 443 202.31.174.9 51205 6 117495 1430
31.65.185.129 443 202.31.174.9 51205 6 117502 1430
133.244.153.246 56862 31.65.185.141 443 6 117506 40
31.65.185.129 443 202.31.174.9 51205 6 117510 1430
222.165.41.192 55767 150.79.7.129 80 6 117514 40
23.234.243.101 80 203.146.254.83 61704 6 117518 52
23.234.243.101 80 203.146.254.83 61702 6 117521 52
23.234.243.99 443 203.146.254.83 61709 6 117525 52
23.234.243.101 80 203.146.254.83 61703 6 117531 52
202.126.14.111 80 150.79.179.98 59791 6 117569 52
150.33.47.65 8932 150.79.7.129 80 6 117576 40
133.227.127.204 53917 103.238.115.79 80 6 117587 40
133.244.153.246 56862 31.65.185.141 443 6 117590 40
23.234.243.99 443 203.146.254.83 61706 6 117603 52
163.39.157.71 52867 199.252.216.15 443 6 117606 436
23.234.243.99 443 203.146.254.83 61707 6 117617 52
163.39.157.71 52867 199.252.216.15 443 6 117633 1426
133.244.153.246 56862 31.65.185.141 443 6 117644 40
133.244.153.246 54194 165.143.250.152 443 6 117694 52
82.102.13.136 1364 133.250.174.99 445 6 117698 48
163.39.157.71 52864 199.252.216.15 443 6 117702 1426
163.39.157.71 52864 199.252.216.15 443 6 117705 436
163.39.157.71 52865 199.252.216.15 443 6 117718 1426
163.39.157.71 52865 199.252.216.15 443 6 117722 436
183.52.183.141 993 202.231.242.67 36931 6 117803 52
133.244.153.246 56862 31.65.185.141 443 6 117809 40
133.244.153.246 56862 31.65.185.141 443 6 117814 40
203.48.9.248 23617 150.79.7.129 80 6 117829 40
64.120.227.69 443 163.39.158.247 58667 6 117869 1426
133.244.153.246 56862 31.65.185.141 443 6 117878 40
131.14.92.245 60000 54.20.141.183 443 6 117881 40
119.125.248.106 10777 133.250.156.245 50356 6 117900 48
133.244.153.246 54194 165.143.250.152 443 6 117968 52
27.178.159.198 4419 150.79.7.129 80 6 117980 40
133.244.153.246 56862 31.65.185.141 443 6 117983 40
64.120.227.69 443 163.39.158.247 58667 6 117994 1426
202.234.224.241 59801 173.199.56.233 80 6 118007 40
125.51.122.124 32415 150.79.7.129 80 6 118016 40
61.211.145.45 61611 150.79.7.129 80 6 118021 40
133.244.153.246 56862 31.65.185.141 443 6 118024 40
203.48.9.248 23620 150.79.7.129 80 6 118043 40
202.234.224.241 59801 173.199.56.233 80 6 118062 40
202.234.224.241 59801 173.199.56.233 80 6 118065 40
23.11.86.235 80 157.210.156.203 47406 6 118069 40
115.109.126.31 42535 150.79.7.129 80 6 118075 40
133.244.153.246 56862 31.65.185.141 443 6 118078 40
114.125.195.70 17294 150.79.7.129 80 6 118092 52
```

```
133.244.153.246 54194 165.143.250.152 443 6 118180 52
133.244.153.246 56862 31.65.185.141 443 6 118193 40
61.43.24.136 80 133.227.178.71 55658 6 118196 1500
163.39.157.71 52864 199.252.216.15 443 6 118201 1426
163.39.157.71 52864 199.252.216.15 443 6 118220 436
110.135.17.73 48692 150.79.7.129 80 6 118420 52
110.135.17.73 48692 150.79.7.129 80 6 118424 52
110.135.17.73 48692 150.79.7.129 80 6 118427 52
150.33.47.65 8932 150.79.7.129 80 6 118439 40
27.178.159.198 4419 150.79.7.129 80 6 118447 40
103.246.81.74 47751 203.146.247.176 80 6 118451 64
133.244.153.246 56862 31.65.185.141 443 6 118491 40
61.243.110.158 10035 150.79.7.129 80 6 118507 40
61.243.110.158 10035 150.79.7.129 80 6 118511 40
157.210.154.200 54684 31.65.191.15 443 6 118514 558
23.225.11.237 80 202.31.174.9 18622 6 118518 1430
133.244.153.246 56862 31.65.185.141 443 6 118521 40
23.225.11.237 80 202.31.174.9 18622 6 118524 1430
23.225.11.237 80 202.31.174.9 18622 6 118566 1430
23.225.11.237 80 202.31.174.9 18622 6 118569 1430
23.225.11.237 80 202.31.174.9 18622 6 118573 1430
23.225.11.237 80 202.31.174.9 18622 6 118576 1430
23.225.11.237 80 202.31.174.9 18622 6 118580 1430
133.244.153.246 56862 31.65.185.141 443 6 118587 40
23.225.11.237 80 202.31.174.9 18622 6 118590 1430
133.244.153.246 54194 165.143.250.152 443 6 118601 52
23.225.11.237 80 202.31.174.9 18622 6 118606 1430
27.178.159.198 4419 150.79.7.129 80 6 118609 40
157.210.145.83 53131 66.36.161.181 443 6 118613 359
23.225.11.237 80 202.31.174.9 18622 6 118616 1430
150.33.47.65 8932 150.79.7.129 80 6 118620 40
150.33.47.65 8932 150.79.7.129 80 6 118623 40
150.33.47.65 8932 150.79.7.129 80 6 118627 40
27.178.159.198 4419 150.79.7.129 80 6 118630 40
27.178.159.198 4419 150.79.7.129 80 6 118635 40
27.178.159.198 4419 150.79.7.129 80 6 118639 40
211.73.188.247 443 202.234.35.13 36955 6 118653 1500
211.73.188.247 443 202.234.35.13 36955 6 118657 303
61.111.37.246 49473 150.79.7.129 80 6 118660 40
118.162.252.133 6095 150.79.7.129 80 6 118665 40
133.244.153.246 56862 31.65.185.141 443 6 118675 40
163.39.157.71 52865 199.252.216.15 443 6 118714 1426
163.39.157.71 52865 199.252.216.15 443 6 118718 436
106.43.9.102 2840 150.79.7.129 80 6 118725 40
221.2.4.133 58658 150.79.7.129 873 6 118729 52
163.39.157.71 52864 199.252.216.15 443 6 118733 436
83.217.137.20 62919 150.79.118.25 2821 6 118739 1458
163.39.157.71 52864 199.252.216.15 443 6 118742 1426
31.65.185.129 443 202.31.174.9 51205 6 118746 1430
```

input:file2.txt

```
12
1 12 13 15 17 21 22 23 27 29 31 36
```

output:Flow.txt

```
0 416.75 1500.00
1 575.00 40.00
2 273.92 931.00
3 20.29 52.00
4 2799.00 40.00
5 316.82 931.00
6 1113.50 52.00
7 304.91 52.00
8 12.00 1344.00
9 119.43 1370.88
10 358.40 1500.00
11 205.86 1500.00
12 331.50 40.00
13 11.00 1500.00
14 268.86 931.00
15 149.67 1500.00
16 201.71 56.50
17 459.80 1300.50
18 451.00 521.00
19 73.03 40.00
20 192.55 1430.00
21 85.33 40.00
22 726.00 40.00
23 6.21 52.00
24 315.17 40.00
25 662.50 1500.00
26 3.00 1500.00
27 26.50 40.00
28 252.00 40.00
29 1303.00 46.00
30 497.00 40.00
31 252.40 1430.00
32 262.75 40.00
33 125.00 1426.00
34 29.00 40.00
35 3.50 52.00
36 4.00 40.00
```

output:KMedoids.txt

```
1635.12
32 4 18 0 13 1 20 25 27 17 6 2
7 12 16 24 28 32
4
18
```

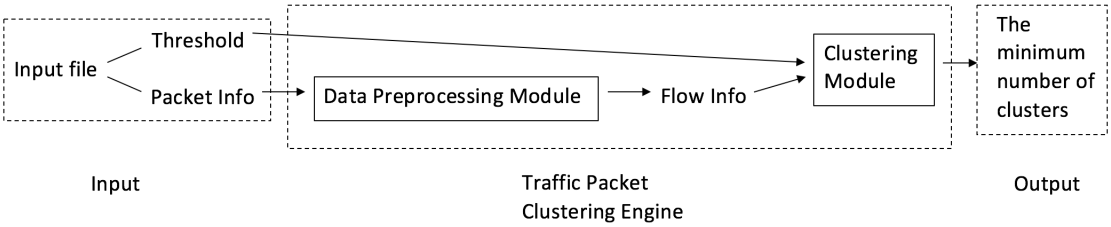
```
0 10
8 13 26
1 22 30
9 11 15 20 31 33
25
3 19 21 23 27 34 35 36
17
6 29
2 5 14
```

output for postgraduate:KMedoidsE.txt

```
1547.02
32 4 18 0 1 13 25 11 2 29 9 27
7 12 16 24 28 32
4
18
0 10 17
1 22 30
13 26
25
11 15 20 31
2 5 14
6 29
8 9 33
3 19 21 23 27 34 35 36
```

1 Background

A data preprocessing module and a clustering module should be implemented, the structure is illustrated below:


Input file
Threshold
Packet Info
Data Preprocessing Module
Flow Info
Clustering Module
The minimum number of clusters
Input
Traffic Packet Clustering Engine
Output