Name: **Kovid Sharma**
zID: **z5240067**

Data Warehousing and Data Mining

Assignment 1

T1, 2020

Table of Contents

Question 1

1.1

	Location	Time	Item	Quantity
0	Melbourne	2005	XBox 360	1700
1	Melbourne	2005	ALL	1700
2	Melbourne	ALL	XBox 360	1700
3	Melbourne	ALL	ALL	1700
4	Sydney	2005	PS2	1400
5	Sydney	2005	ALL	1400
6	Sydney	2006	PS2	1500
7	Sydney	2006	Wii	500
8	Sydney	2006	ALL	2000
9	Sydney	ALL	PS2	2900
10	Sydney	ALL	Wii	500
11	Sydney	ALL	ALL	3400
12	ALL	2005	PS2	1400
13	ALL	2005	XBox 360	1700
14	ALL	2005	ALL	3100
15	ALL	2006	PS2	1500
16	ALL	2006	Wii	500
17	ALL	2006	ALL	2000
18	ALL	ALL	PS2	2900
19	ALL	ALL	Wii	500
20	ALL	ALL	XBox 360	1700
21	ALL	ALL	ALL	5100

1.2

```
SELECT *
FROM [Location] CROSS JOIN [Time] CROSS JOIN [ITEM]
```

1.3

	Location	Time	Item	Quantity
1	Sydney	2006	ALL	2000.0
2	Sydney	ALL	PS2	2900.0
3	Sydney	ALL	ALL	3400.0
4	ALL	2005	ALL	3100.0
5	ALL	2006	ALL	2000.0
6	ALL	ALL	PS2	2900.0
7	ALL	ALL	ALL	5100.0

1.4

$$f(\text{Location},\text{Time},\text{Item}) = 4^2 \cdot f(\text{Location}) + 4 \cdot f(\text{Time}) + f(\text{Item})$$

ArrayIndex	Value
21	1400
25	1500
27	500
38	1700

Question 2

	P1	P2	P3	P4	P5
P1	1	0.10	0.41	0.55	0.35
P2	0.10	1	0.64	0.47	0.98
P3	0.41	0.64	1	0.44	0.85
P4	0.55	0.47	0.44	1	0.76
P5	0.35	0.98	0.85	0.76	1

Classes = [{1}, {2}, {3}, {4}, {5}]

Iteration 1

	P1	P2	P3	P4	P5
P1	1	0.10	0.41	0.55	0.35
P2	0.10	1	0.64	0.47	0.98
P3	0.41	0.64	1	0.44	0.85
P4	0.55	0.47	0.44	1	0.76
P5	0.35	0.98	0.85	0.76	1

Max is 0.98 = (2,5)

Classes = [{1}, {2,5}, {3}, {4}]

Iteration 2

	P1	P 2,5	P3	P4
P1	1	0.477	0.41	0.55
P 2,5	0.477	1	0.823	0.736
P3	0.41	0.823	1	0.44
P4	0.55	0.736	0.44	1

Max is 0.823 = ((2,5), 3)

Classes = [{1}, {2,3,5}, {4}]

Iteration 3

	P1	P 2,5,3	P4
P1	1	0.555	0.55
P 2,5,3	0.555	1	0.69
P4	0.55	0.69	1

Max is 0.69 = ((2,5,3), 4)

Classes = [{1}, {2,3,4,5}]

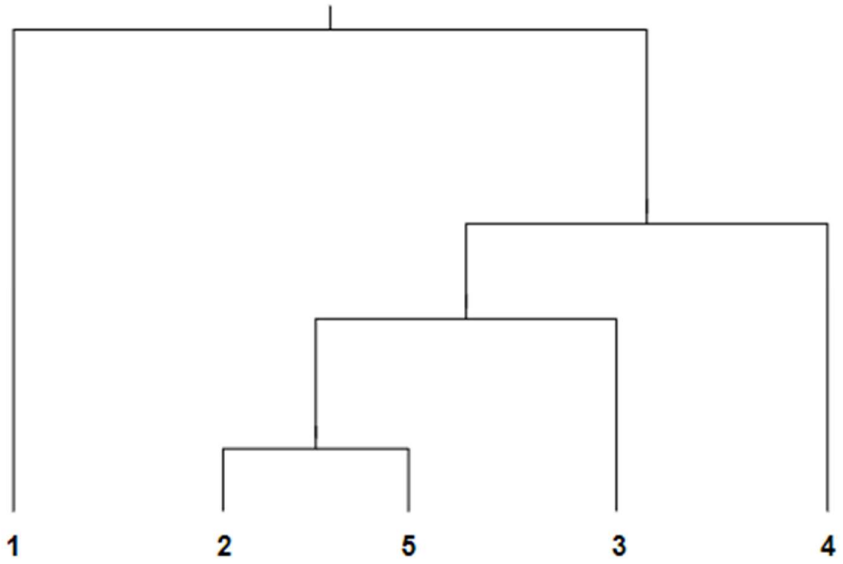
Iteration 4

	P1	P 2,5,3,4
P1	1	0.555
P 2,5,3,4	0.555	1

Max is 0.555 = ((2,5,3,4), 1)

Classes = [{1,2,3,4,5}]

Dendrogram:



Question 3

3.1

Add this function after line 8.
canStop ← IsCenterChange(C;G)

Algorithm 1: *k-means(D,k)*

Require: C set of k centers, G set of clusters
function IsCenterChange*(C, G)*
for all $g \in G$ **do**
$temp_i \leftarrow$ ComputeCenter(g)
if $c_i \neq temp_i$ **then**
return false
end if
end for
return true
end function

3.2

After calculating the center, the new center is the minimum point of $\Sigma_{i=0}^{n}$ *Cost(gi)* (by definition of center point).

After finding the nearest center, if the distance from a point to current cluster center is larger than another center, this point will move to the other center. So, at this step, the total distance will decrease too (otherwise the point will stay in current cluster set).

Combined with these two steps, the total distance will decrease too.

3.3

Using conclusion of 3.2, we know the total cost(distance) of clustering will never increase.

1. If the old clustering is the same as the new, then the next clustering will again be the same.
2. If the new clustering is different from the old then the newer one has a lower cost.

Total number of possible clusters is k^N .
k is the number of clusters.
N is number of entries.
So, the loop is finite. Until all cluster get their local minimum, the loop will be end.
So, it always converges to a local minimum.