9318 assignment

Lipeng TAO
Z5048267

Q1

1，

Location	Time	Item	Sum(quantity)
Mel	2005	XBOX360	1700
Mel	2005	ALL	1700
Mel	ALL	XBOX360	1700
Mel	ALL	ALL	1700
Syd	2005	PS2	1400
Syd	2005	ALL	1400
Syd	2006	PS2	1500
Syd	2006	Wii	500
Syd	2006	ALL	2000
Syd	ALL	PS2	2900
Syd	ALL	Wii	500
Syd	ALL	ALL	3400
ALL	2005	PS2	1400
ALL	2005	XBOX360	1700
ALL	2005	ALL	3100
ALL	2006	Wii	500
ALL	2006	PS2	1500
ALL	2006	ALL	2000
ALL	ALL	PS2	2900
ALL	ALL	XBOX360	1700
ALL	ALL	Wii	500
ALL	ALL	ALL	5100

2，

```
（1） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
     GROUP BY Location , Time , Item
     WITH ROLLUP
     UNION
（2） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
     GROUP BY Location , Item , Time
     WITH ROLLUP
     UNION
（3） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
     GROUP BY Time , Location , Item
     WITH ROLLUP
     UNION
（4） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
```

```
    GROUP BY Time , Item , Location
    WITH ROLLUP
    UNION
（5） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
    GROUP BY Item , Time , Location
    WITH ROLLUP
    UNION
（6） SELECT Location , Time , Item , SUM( Quantity ) FROM Sales
    GROUP BY Item , Location , Time
    WITH ROLLUP
    UNION
```

3,

Location	Time	Item	Sum(quantity)
Syd	2006	ALL	2000
Syd	ALL	PS2	2900
Syd	ALL	ALL	3400
ALL	2005	ALL	3100
ALL	2006	ALL	2000
ALL	ALL	PS2	2900
ALL	ALL	ALL	5100

4,

The mapping function is: $f(Location, Time, Item) = (3 * 4) * Location + 4 * Time + Item$.

And the index range is from 0 to 38.

Q2

1, Cluster p2 and p5

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

2, Cluster p25 and p3

	P1	P25	P3	P4
P1		0.35	0.41	0.55
P25	0.35		0.85	0.76
P3	0.41	0.85		0.44
P4	0.55	0.76	0.44	

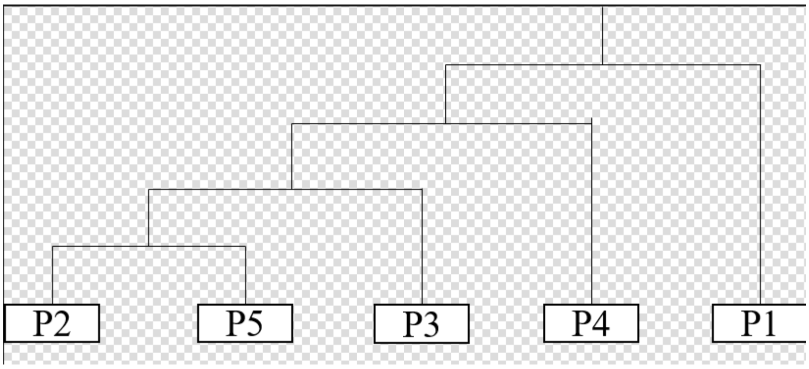
3, Cluster p253 and p4

	P1	P253	P4
P1		0.41	0.55
P253	0.41		0.76
P4	0.55	0.76	

4, Cluster p2534 and p1

	P1	P2534
P1		0.41
P2534	0.41	

The cluster`s result as below:



Q3

1, As the code shown below, I add 3 new lines, line 9 and line 13-14, which sign up with red color

```
Initialize $k$ centers $C = [c_1, c_2, . . . , c_k]$;
$canStop$ ← False
while $canStop$ = False do
    Initialize $k$ empty clusters $G = [g_1, g_2, . . . , g_k]$;
    for each data point $p \in D$ do
        $c_x$ ← NearestCenter($p,C$);
        $g_{c_x}$.append($p$);
    end for
    $tmpCenter$ ← $C$
    for each group $g \in G$ do
        $c_i$ ← ComputeCenter($g$);
    end for
    if $lastCenter = C$ then
        $canStop$ ← True
    end if
end while
return $G$
```

2,

Total cost is the sum of distances between their original points and their center points. The Kmeans algorithm will be divided in two steps

(1), points are attached to the nearest center. Assume that point p is attached in center ci and it will be attached to cj after this step. According to the algorithm, the distance between p and ci is smaller than p and cj . So, in this step, the cost does not increase.

(2), new centers are computed in new clusters. The point has the smallest sum of distance between each point in this cluster becomes the new center. So, $\Sigma p \in gi\ dist2(p, newci) \leq \Sigma p \in gi\ dist2(p, oldci)$, where new c stands for the centers after step 2 while old c stands for the center before step 2. The cost does not increase in step 2.

Therefore, the total cost will not increase in Kmeans algorithm

3, According to the conclusion in (2), the cost does not increase in kmeans algorithm. Therefore, the cost keep same or it decreases in every iteration. The points cannot find a better way from cluster and the centers already are the nearest centers when the algorithm`s flow finish. So this algorithm converges to a local minimize