COMP9318 (20T1) ASSIGNMENT 1

DUE ON 20:59 12 APR, 2020 (SUN)

Q1. (*40 marks*)

Consider the following base cuboid *Sales* with *four* tuples and the aggregate function SUM:

Location	*Time*	*Item*	*Quantity*
Sydney	2005	PS2	1400
Sydney	2006	PS2	1500
Sydney	2006	Wii	500
Melbourne	2005	XBox 360	1700

Location, *Time*, and *Item* are dimensions and *Quantity* is the measure. Suppose the system has built-in support for the value **ALL**.

(1) List the tuples in the complete data cube of R in a tabular form with 4 attributes, i.e., $Location, Time, Item, \text{SUM}(Quantity)$?

(2) Write down an equivalent SQL statement that computes the same result (i.e., the cube). You can *only* use standard SQL constructs, i.e., no **CUBE BY** clause.

(3) Consider the following *ice-berg cube* query:

```
SELECT  Location, Time, Item, SUM(Quantity)
FROM    Sales
CUBE BY Location, Time, Item
HAVING  COUNT(*) > 1
```

Draw the result of the query in a tabular form.

(4) Assume that we adopt a MOLAP architecture to store the full data cube of R, with the following mapping functions:

$$f_{Location}(x) = \begin{cases} 1 & \text{if } x = \text{'Sydney'}, \\ 2 & \text{if } x = \text{'Melbourne'}, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

$$f_{Time}(x) = \begin{cases} 1 & \text{if } x = 2005, \\ 2 & \text{if } x = 2006, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

$$f_{Item}(x) = \begin{cases} 1 & \text{if } x = \text{'PS2'}, \\ 2 & \text{if } x = \text{'XBox 360'}, \\ 3 & \text{if } x = \text{'Wii'}, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

Draw the MOLAP cube (i.e., sparse multi-dimensional array) in a tabular form of $(ArrayIndex, Value)$. You also need to write down the function you chose to map a multi-dimensional point to a one-dimensioinal point.

Q2. (*30 marks*)

Consider the given **similarity** matrix. You are asked to perform group average hierarchical clustering on this dataset.

You need to show the steps and final result of the clustering algorithm. You will show the final results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged.

	p_1	p_2	p_3	p_4	p_5
p_1	1.00	0.10	0.41	0.55	0.35
p_2	0.10	1.00	0.64	0.47	0.98
p_3	0.41	0.64	1.00	0.44	0.85
p_4	0.55	0.47	0.44	1.00	0.76
p_5	0.35	0.98	0.85	0.76	1.00

Q3. (*30 marks*)

Algorithm 1: k-means(D, k)

Data: D is a dataset of n d-dimensional points; k is the number of clusters.

```
1  Initialize k centers C = [c_1, c_2, ..., c_k];
2  canStop ← false;
3  while canStop = false do
4      Initialize k empty clusters G = [g_1, g_2, ..., g_k];
5      for each data point p ∈ D do
6          c_x ← NearestCenter(p, C);
7          g_{c_x}.append(p);
8      for each group g ∈ G do
9          c_i ← ComputeCenter(g);
10 return G;
```

Consider the (slightly incomplete) k-means clustering algorithm as depicted in Algorithm 1.

(1) Assume that the stopping criterion is till the algorithm converges to the final k clusters. Can you insert several lines of pseudo-code to the algorithm to implement this logic? You are **not** allowed to change the first 7 lines though.
(2) The cost of k clusters is just the total cost of each group g_i, or formally

$$cost(g_1, g_2, \ldots, g_k) = \sum_{i=1}^{k} cost(g_i)$$

$cost(g_i)$ is the sum of squared distances of all its constituent points to the center c_i, or

$$cost(g_i) = \sum_{p \in g_i} dist^2(p, c_i)$$

$dist()$ is the Euclidean distance. Now show that the cost of k clusters as evaluated at the end of each iteration (i.e., after Line 9 in the current algorithm) never increases. (You may assume $d = 2$)
(3) Prove that the cost of clusters obtained by k-means algorithm always converges to a local minima. You can make use of the previous conclusion even if you have not proved it.

Hint 1. *In fact, the two loops (Lines 5–7 and Lines 8–9) never increases the cost.*

Submission

Please write down your answers in a file named `ass1.pdf`. You **must write down your name and student ID on the first page**. You should typeset your answers in LaTeX or MS Word. We do **not** accept handwritten answers.

You can submit your file by

```
give cs9318 ass1 ass1.pdf
```

Late Penalty. -10% per day for the first two days, and -20% for each of the following days.