1.A. Step 1, $L_0 = \phi$, $k=1$, $C_1 = \{\{1\}, \{2\} \ldots, \{16\}, \{19\}\} - \{14\}$

MFCS $= \{1, 4, \ldots 16, 19\} - \{14\}$, MFS $= \phi$

Counts:

$\{1\} = 2$, $\{2\} = 3$, $\{3\} = 4$, $\{4\} = 1$, $\{5\} = 1$

$\{6\} = 4$, $\{7\} = 1$, $\{8\} = 1$, $\{9\} = 1$, $\{10\} = 1$, $\{11\} = 1$

$\{12\} = 2$, $\{13\} = 3$, $\{15\} = 2$, $\{16\} = 3$, $\{19\} = 1$

MFCS $= \{1, 2, \cdots 16, 19\} - \{14\} = 0$

MFS $= \phi$

$S_1 = \{4, 5, 7, 8, 9, 10, 11, 19\}$

$L_1 = \{1, 2, 3, 6, 12, 13, 15, 16\}$

Since $S_1 \neq \phi$, call MFCS-gen algo

$S = \{4\}$

$m = \{1, 2, \cdots 16, 19\} - \{14\}$

Removing S from m, $\Rightarrow$ $\{1, 2, 3, 5 \cdots 16, 19\} - \{14\}$

Similarly we remove $\langle 5, 7, 8, 9, 10, 11, 19 \rangle$

$\Rightarrow$ MFCS $= \langle 1, 2, 3, 6, 12, 13, 15, 16\}$

MFS $= \phi$

Now we generate C_2 from L_1

$$C_2 = \{\{1,2\}, \{1,3\}, \{1,6\}, \{1,12\}, \{1,13\}, \{1,15\},$$
$$\ldots \quad \{15,16\}\}$$

Now we call MFCS-prune to prune candidates in C_2.

Since all items of C_2 is a subset of MFCS, nothing is pruned.

K++.

Just like for $K=1$, for $K=2$ we calculate the count for all data in C_2 and form S_2 & L_2 and then we calculate C_3. We keep doing this till $C_K = \phi$

B. Let a frequent itemset $= S$

Let $m \subseteq S$ (subset), $m \neq \phi$.

We know wherever S occurs, m also occurs, and maybe even at other places.
So count of $m \geq$ count of S.

count of $S \geq$ min support, because S is frequent itemset.

$\therefore$ count of $m \geq$ ~~min support~~ count of $S \geq$ min support

c. False. Example: Let 2 products be laptop & mouse.

laptop $\rightarrow$ mouse but mouse $\not\rightarrow$ laptop.

Every person who buys a laptop will buy a mouse, but mouse can be bought by a person who is not buying laptop.

Q2. A. To find out which attribute has to be branched before the others, we find the gain for the various attributes.

Let D be the entire database

$$\text{info}(D) = -\frac{5}{10}\log(5/10) - 5/10\log(5/10) = 1$$

$$\text{info}_{\text{colour}}(D) = \frac{5}{10}\left(-\frac{3}{5}\log(3/5) - \frac{2}{5}\log(2/5)\right)$$
$$+\frac{5}{10}\left(-\frac{3}{5}\log(3/5) - \frac{2}{5}\log(2/5)\right)$$
$$= 0.966$$

$$\text{info}_{\text{type}}(D) = \frac{6}{10}\left(-\frac{4}{6}\log(4/6) - \frac{2}{6}\log(2/6)\right)$$
$$+\frac{4}{10}\left(-3/4\log(3/4) - \frac{1}{4}\log(1/4)\right)$$
$$= 0.866$$

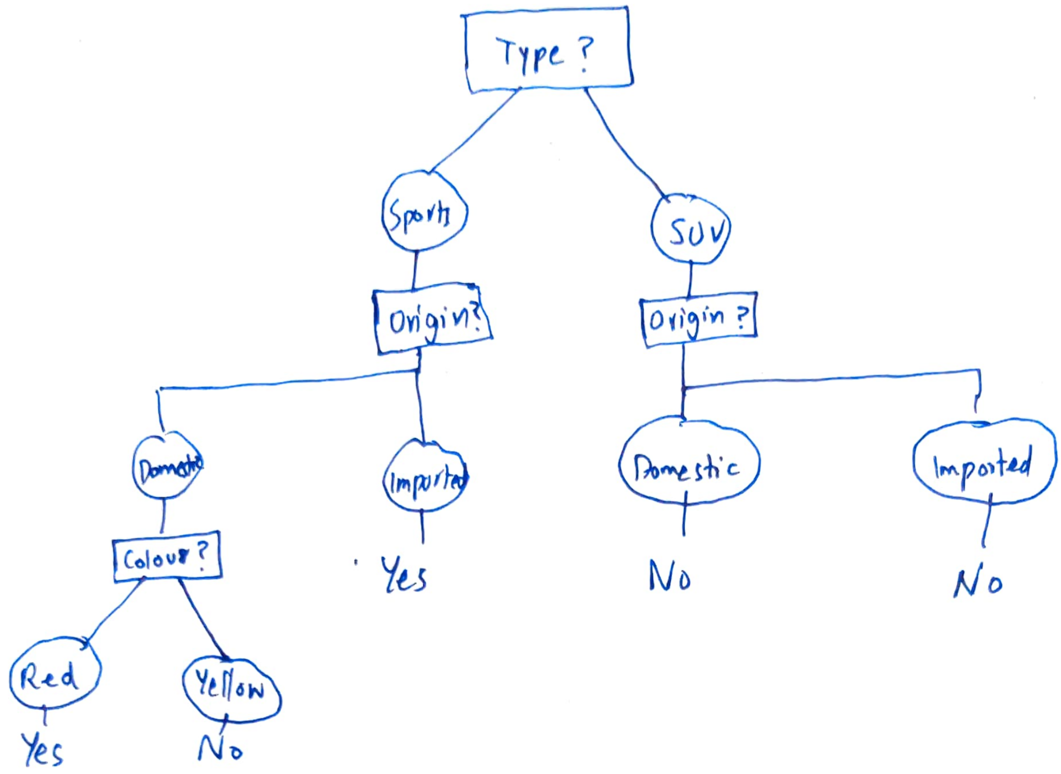
$$\text{info}_{\text{origin}}(D) = \frac{5}{10}\left(-3/5\log(3/5) - \frac{2}{5}\log(2/5)\right)$$
$$+\frac{5}{10}\left(-3/5\log(3/5) - \frac{2}{5}\log(2/5)\right)$$
$$= 0.966$$

$$\text{Gain}(\text{color}) = 1 - 0.966 = 0.034$$

$$\text{Gain}(\text{type}) = 1 - 0.866 = 0.134$$

$$\text{Gain}(\text{origin}) = 1 - 0.966 = 0.034$$

Thus branching using the "type" attribute gives maximum information gain.



<Sports, Domestic, Red> has 2 "Yes" and 1 "No";
Using majority voting, it is made to "Yes"

<SUV, Imported, *> has 2 "No" and 1 "Yes";
Using majority voting, it is made to "No".

2.B

$$h(x) = h(x_1, x_2) = \begin{cases} 1, & \text{if } (-12 + 8(x_1 + x_2)) > 0 \\ 0, & \text{if } (-12 + 8(x_1 + x_2)) \leq 0 \end{cases}$$

3A.

1 → (2,10)

2 → (2,5)

3 → (8,4)

4 → (5,8)

5 → (7,5)

6 → (6,4)

7 → (1,2)

8 → (4,9)

9 → (8,6)

10 → (6,7)

Initial centroids → (2,10), (2,5), (8,4)

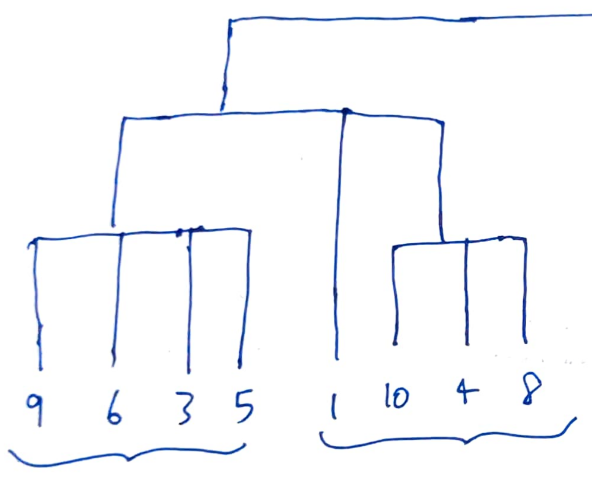
	Cluster 1		Cluster 2		Cluster 3	
Step	Individual	Mean	Individual	Mean	Individual	Mean
1	1	(2,10)	-	(2,5)	-	(8,4)
2	1	(2,10)	(2)	(2,5)	-	(8,4)
3	1	(2,10)	(2)	(2,5)	(3)	(8,4)
4	1	(2,10)	(2,4)	(3.5,6.5)	(3)	(8,4)
5	1	(2,10)	(2,4)	(3.5,6.5)	(3,5)	(7.5,4.5)
6	1	(2,10)	(2,4)	(3.5,6.5)	(3,5,6)	(7,4.3)
7	1	(2,10)	(2,4,7)	(2.6,5)	(3,5,6)	(7,4.3)
8	1,8	(3,9.5)	(2,4,7)	(2.6,5)	(3,5,6)	(7,4.3)
9	1,8	(3,9.5)	(2,4,7)	(2.6,5)	(3,5,6,9)	(7.25,4.75)
10	1,8	(3,9.5)	(2,4,7)	(2.6,5)	(3,5,6,9,10)	(7,5.2)

The 3 clusters are, with centroids:

No.	Points	centroids
1.	<1,8>	(3,9.5)
2.	<2,4,7>	(2.6,5)
3.	<3,5,6,9,10>	(7,5.2)

B. For hierarchical clustering first we create the 10x10 distance (euclidean) matrix. Find the smallest element, which gives the pair of closest points. We ~~group~~ merge them to same point and repeat.

After repeating this for 9 iterations, we have the following structure.



We can cluster in the following way.

Cluster 1 → <9,6,3,5>

Cluster 2 → <1,10,4,8>

Cluster 3 → <2,7>

This is almost giving similar results as k-means clustering output.

Time complexity:

1) K-Means Clustering:

Let there be 'n' points, & 'k' clusters.

Each point has to be checked on all 'k' clusters;

~~a~~ Hence time complexity = $O(n \cdot k)$

2) Hierarchical clustering:

At each step the matrix dimension of $n \times n$ becomes $(n-1) \times (n-1)$, till it reaches 1×1.

Assuming a brute force approach, time complexity

$= n^2 + \cancel{(n-1) \times (n-1)} (n-1)^2 + (n-2)^2 \dots 1^2$

$= O(n^3)$

where n is the number of points.

4. A.

Precision = True positive /(true positive + false positive)

Recall = True positive (True positive + false negatives)

True positives = 5

False positives = 15

False negatives = 5

Precision = $5/(5+15) = 5/20 = 1/4 = 25\%$

~~b~~. Recall = $5/(5+5) = 5/10 = 1/2 = 50\%$

B.

a. Clearly the 75^{th} percentile is weight = 23.

Hence ~~s~~he is wrong.

b. Median = 17. IQR = 7

c. Total bags = 240, less than 10 represent one-forth of total. Hence 60 bags with weight less than 10.