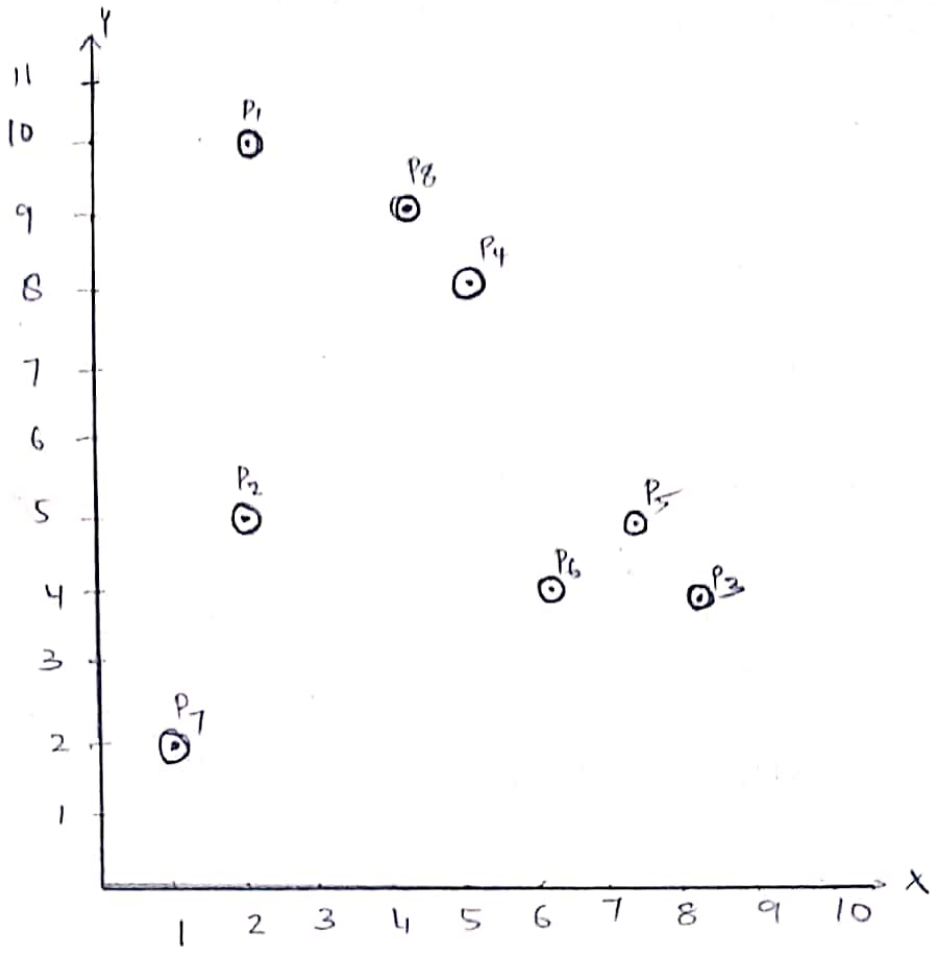
1. a.



b. 3 clusters

C_1 : $\{P_7, P_2\}$ C_2: $\{P_1, P_8, P_4\}$ C_3: $\{P_5, P_6, P_3\}$

c. Centres : C_1 $(2,5)$ C_2 $(5,8)$ C_3 $(4,9)$

	Point	Distance C_1	Distance C_2	Distance C_3	Belongs to
P_1	(2,10)	5	$\sqrt{13}$	$\sqrt{5}$	C_3
P_2	(2,5)	0	-	-	C_1
P_3	(8,4)	$\sqrt{37}$	5	$\sqrt{41}$	C_2
P_4	(5,8)	-	0	-	C_2
P_5	(7,5)	5	$\sqrt{13}$	5	C_2
P_6	(6,4)	$\sqrt{17}$	$\sqrt{17}$	$\sqrt{29}$	C_1
P_7	(1,2)	$\sqrt{10}$	$\sqrt{52}$	$\sqrt{58}$	C_1
P_8	(4,9)	-	-	0	C_3

C_1: $\{P_2, P_6, P_7\}$

C_2: $\{P_3, P_4, P_5\}$

C_3: $\{P_1, P_8\}$

d) Centre of cluster after 1st iteration :

Cluster 1 : (P_2, P_6, P_7)

Cluster 2 : (P_3, P_4, P_5)

Cluster 3 : (P_1, P_8)

Center 1 : $\frac{2+6+1}{3}, \frac{5+4+2}{3} = 3, 3.66$

Center 2 : $\frac{8+5+7}{3}, \frac{4+8+5}{3} = 6.66, 5.66$

Center 3 : $\frac{2+4}{2}, \frac{10+9}{2} = 3, 9.5$

C1 center : $\{3, 3.66\}$

C2 center : $\{6.66, 5.66\}$

C3 center : $\{3, 9.5\}$

e. Center of cluster after 2nd iteration :

$P_1 - C_3$, $P_2 - C_1$, $P_3 - C_2$, $P_4 - C_3$, $P_5 - C_2$, $P_6 - C_2$
$P_7 - C_1$, $P_8 - C_3$

~~Cluster 1 : $\{P_2, P_7\}$ Cluster 2 : $\{P_3, P_4, P_5\}$ Cluster 3 : $\{$~~

Cluster 1 : $\{P_2, P_7\}$

Cluster 2 : $\{P_3, P_5, P_6\}$

Cluster 3 : $\{P_1, P_4, P_8\}$

C_1: $\frac{3}{2}, \frac{7}{2}$ = $(1.5, 3.5)$

C_2: $\frac{8+7+6}{3}, \frac{4+5+4}{3}$ = $\left(7, \frac{13}{3}\right)$

C_3: $\frac{2+5+4}{3}, \frac{10+8+9}{3}$ = $\left(\frac{11}{3}, 9\right)$

f. $P_1 - C3$, $P_2 - C1$, $P_3 - C2$, $P_4 - C3$, $P_5 - C2$, $P_6 - C2$, $P_7 - C1$, $P_8 - C3$

Cluster 1: $\{P_2, P_7\}$

Cluster 2: $\{P_3, P_5, P_6\}$

Cluster 3: $\{P_1, P_4, P_8\}$

Center after 3rd iteration:
Same as 2nd iteration centres.

g. The results are similar to what was guessed in section b.

h. 3 iterations

i. Resulting centres: $(1.5, 3.5)$, $\left(7, \frac{13}{3}\right)$, $\left(\frac{11}{3}, 9\right)$

Resulting Clusters:

Cluster 1: $\{P_2, P_7\}$

Cluster 2: $\{P_3, P_5, P_6\}$

Cluster 3: $\{P_1, P_4, P_8\}$

2

Singe Link clustering

P_2, P_5 share high similarity. Merge P_2 P_5

	P_1	P_3	P_4	P_2P_5
P_1	1.0	0.41	0.55	0.35
P_3	0.41	1.0	0.44	0.85
P_4	0.55	0.44	1.0	0.76
P_2P_5	0.35	0.85	0.76	1.0

P_3 P_2P_5 share highest similarity Merge P_3, P_2P_5

	P_1	P_4	$P_3P_2P_5$
P_1	1.0	0.55	0.41
P_4	0.55	1.0	0.76
$P_3P_2P_5$	0.41	0.76	1.0

Merge P_4, $P_3P_2P_5$ (0.76)

	P_1	$P_4P_3P_2P_5$
P_1	1.0	0.55
$P_4P_3P_2P_5$	0.55	1.0

Complete Link Clustering

P_2 P_5 share high similarity
Merge P_2P_5

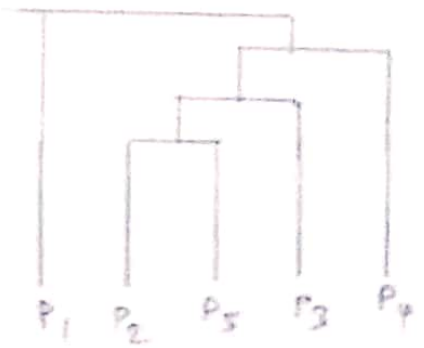
	P_1	P_3	P_4	P_2P_5
P_1	1.0	0.41	0.55	0.10
P_3	0.41	1.0	0.44	0.64
P_4	0.55	0.44	1.0	0.47
P_2P_5	0.10	0.64	0.47	1.0

P_3, P_2P_5 share high sim.
Merge P_3, P_2P_5

	P_1	P_4	$P_3P_2P_5$
P_1	1.0	0.55	0.10
P_4	0.55	1.0	0.44
$P_3P_2P_5$	0.10	0.44	1.0

P_4, $P_3P_2P_5$ share high sim. Merge P_4, $P_3P_2P_5$

	P_1	$P_4P_3P_2P_5$
P_1	1.0	0.10
$P_4P_3P_2P_5$	0.10	1.0



Both the methods merge the points in a similar way as we see that the above two dendograms are same.

3. Epsilon = 2 Min_Samples = 2

a. There are three clusters discovered.

Cluster 1: (2,10) (2,5) (1,2)

Cluster 2: (8,4) (7,5) (6,4)

Cluster 3: (5,8) (4,9)

b. Epsilon = $\sqrt{10}$

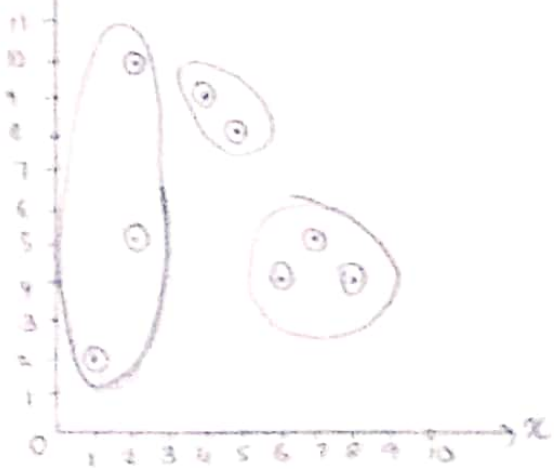
There are three clusters

Cluster 1: (2,10), (5,8), (4,9)

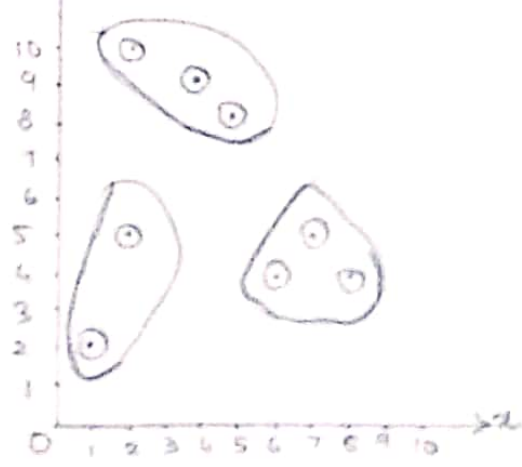
Cluster 2: (2,5), (1,2)

Cluster 3: (8,4), (7,5), (6,4)

a) Epsilon = 2



b) Epsilon = $\sqrt{10}$



PART #2

Q1.

	Cassandra	Big Table
i)	Cassandra belongs to "Database" Category of tech stack	Big Table Classified as "NoSQL Database as a Service"
ii)	Primary reason : "Distributed"	Primary Reason : "High performance"
iii)	A partitioned row store. Rows are organised into tables with a required primary key.	A fast, fully managed, scalable NoSQL database service ideal for web, Mobile, IoT Apps requiring TB to PB of data.
iv)	Cassandra automatically distributes data across multiple nachins as Cluster size changes	Bigtable has been widely used in Google Analytics & Gmail.

Q2. Apache cassandra is a distributed DBMS that is built to handle large amounts of data across multiple data centres and the Cloud.

Key features :

→ Highly Scalable
→ offers high availability
→ Has no single point of failure.

It is a NoSQL DB meaning DB stores & retrieves data without requiring data to be stored in Tabular format

Q3. Tunable Consistency in Cassandra:

Apache Cassandra is a "AP" system which means it prefers data availability over consistency. To ensure data availability, the data updates should be propagated across networks to remote hosts. If the hosts are down, it may take time to update the data & users may read stale/ not up to data information. To avoid this Tunable Consistency is used.

When performing a read/write operation a database client can specify a consistency level. The consistency level refers to the no. of replicas that need to respond for a read or write operation to be considered complete.

For a less important data, it is set to ONE,
For accuracy driven data, TWO, THREE or QUORUM.

Q4. Memtable:

When a write occurs, Cassandra stores the data in a memory structure called Memtable.
The Memtable is a write-back cache of data partitions that cassandra looks up by key.
The Memtable stores writes in sorted order until reaching a configurable limit, & then it is flushed.

Q5. SStables are the immutable data files that Cassandra uses for persisting data on disk.

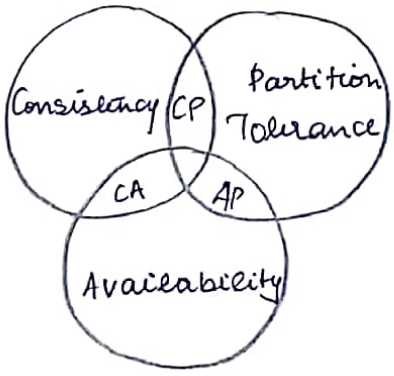
When SStables are flushed to disk from Memtables or are streamed from other nodes, Cassandra triggers compactions which combine multiple SS tables into one.

SS - Sorting String Table is a file of key/value string pairs, sorted by keys.

SStable - Immutable, Relational table - can't be updated, edited & so on.

Q6. CAP theorem (Brewer's theorem)

CAP theorem states that a distributed system can only guarantee two out of these three characteristics: Consistency, Availability and Partition Tolerance.



Q7. A tablet server stores and serves tablets to clients. For a given tablet, one tablet server acts as a leader and the others serve follower repeicas of that tablet.

One tablet server can serve multiple tablets, and one tablet can be served by multiple tablet servers

A tablet is a contiguous segment of a table, similar to a partition in other data storage engines or relational databases.

A given tablet is replicated on multiple tablet servers and at a given pt in time, one of these replicas is considered the leader tablet.

Any replicas can service reads. Writes require consensus among set of tablet servers serving the tablet.