**Student Name:**

**Paper 1 [Hadoop++]**

[5 Points] Q1: In Hadoop++, the Trojan index is created at which granularity? Is it global (file-level) or local (block-level)? *[Just mention the granularity]*

[5 Points] Q2: Describe the layout of a Trojan Join index over two files R and S? That is, how the data is pre-processed, organized on HDFS, and then joined at query time.

*[Draw (plot) the layout, 3-4 sentences how the data is pre-processed to generate the layout, 2-3 sentences descripting the join job at query time]*

[5 Points] Q3: How many indexes Hadoop++ can create for a given dataset? How many of those can be “clustered” vs. “un-clustered”? *[Justify your answer in few 3-4 sentences]*

[5 Points] Q4: If we have a Trojan index on column X of a file that consists of 200 HDFS blocks, and we have a selection query that selects all records where column X = *100.* Assume that value 100 in column X appears only in 10 blocks of the file. Answer the following:

1. Do we need a map-only job or map-reduce job to process the above query?
2. How many mappers will get started in Hadoop++ as part of the above query? *[Just Number]*
3. How the index will save some work  *[3-4 sentences]*

**Paper 2 [Simba]**

[5 Points] Q1: A key contribution in Simba is *Indexing.* Given the following RDD and its partitions, show the structure of the index that Simba builds on such RDD *(that is: complete the figure to show the structure of the index)*



[5 Points] Q2: For KNN query, you are given a query point (indicated as X) in the figure below. Assume the points in the space are partitioned as illustrated in the figure, and each partition has an ID, e.g., P1, P2, …



Write down the steps that Simba will take to compute the KNN for the given query point under two cases (indicate what search boundaries will be set and which partitions will be touched):

Case 1: K = 2 // *You can put marks on the figure to help you referencing things*



Case 2: K = 4 *//You can put marks on the figure to help you referencing things*

