Ans 1.In the first part of question when we have to execute text based queries we will create a TF & IDF table using PDF names as document id and we will create a Inverted index for that. After that using inverted index we can execute text based queries and return the results with rankings on the basis of cosine similarity score.

Second Part we need execute queries Text + Spatial locations. So what we do is create a KD-Tree on the basis of locations of objects provide to us. We have to make sure that the generated KD-tree should be balanced and accurate on the basis of location.

After creating KD-Tree of the given documents we try execute text + spatial queries.

When we need to execute Text + Spatial queries firstly we execute simple text queries and arrange the results on the basis of cosine similarities. After that we use spatial location in query and try to execute a point query on KD-Tree and find K nearest neighbors.

When we get K nearest neighbors after that we assign score to each of the documents based on their distance from query spatial location point. Highest score to closest neighbor and lowest score to furthest neighbor. Then we add both scores of document and rank the documents according to the new score. But firstly we normalize scores .4\*(cosine similarity) + .6\*(distance based score).

Third Part: If a single document is geo-tagged with multiple locations than this scheme can be extended to that, if the similar document with multiple locations comes in nearest neighbor than its score of distance will be added twice and it would get greater preference compared to other document.