**Design Document For Prefix matching Problem:**

**Problem Statement:**

To Suggest the top 10 names from the list of name score pairs that match a query string given by a user. We will Consider a name to be a match for a query if:

* The query is the prefix of the name
* The query is a prefix of a part of the name following an underscore.

**Solution**:

**Input Type:** In this code we are assuming that the input will be a file containing list of tuples. Each tuple will have a name ,score pair - > (name,score)

**Output Type**: We will print a list containing top 10 names. We also save the output in output.json file which contains the json object having key as the query and result as the value so that we don’t need to process the file again and again for the same query input.

**Logic :**

1. Search for the query in output.json file, it it exists then return the result as we don’t need to process the file again.
2. If we don’t get any result in 1st step we start processing the file. As our file may contain millions of records and we have limited memory we create a file pointer and read the data in chunks by defining the chunk size based on our machines configuration. We take each chunk and convert them into a list of tuples. We take that list and maintain the top 10 names by creating a min heap of 10 name,score pairs where score acts as the key.

Once we complete processing all the chunks we convert that heap into the dictionary with query as the key and top 10 names as the value and print the top 10 names as the output.

1. Once we get the output after fileprocessing we create a output.json file if doesn’t exists and add the json object saving the result for the query. If file already exists then we update the json file by adding result of the current query as well

**Alternate Approach:**

We can have use multiple approaches for solving this issue, one more approach that we can think of is to read the file in chunks and keep saving the name, score pairs in db indexing it on score so that at the end we can fetch the top 10 names based on score, but the issue with this approach is that we will have to save the entire data in db.

**Improvements:**

We didn’t introduce any logging mechanism in this project. We can have a log file logging the required details.