**Question**

Duplicate URLs:

You have 10 billion URLs. How do you detect the duplicate documents? In this case, assume “duplicate” means that URLs are identical.

**Solution**

1. When we are going to detect 10 billion URL’s if we are going to compare each URL with each other like for loop it will be big (n2) and it is a bad design.
2. We could make use of Hashing concept to solve the same. i.e., take a single for loop big (n) and get the hash value and check if we have the URL present in the location. If not then we can add the URL and proceed with the next. In this way we could avoid big (n2) complexity using big (n) instead.
   1. While a hashing concept is used we need to consider the following
      1. How much hash indices are allowed. If we have a server space of 10 billion URL capacity then we can index it is each location uniquely and then there is a big (1) unit time when trying to check if the Duplicate URL exists. Now if there is no URL in that location then no duplicates exists and so we can insert the record and proceed with the next.
      2. If the space is restricted then we can leverage the linked list concept to store the URL based on their hash values. If two different URL’s are having the same hash then we add them in a form of linked list. This will however slightly be slow since if many URLs are having a same hash index then it might be taking some time to navigate to each linked list and verify.
3. Since it has not been given what needs to be made with the URL here, whether to proceed finding all the duplicates and listing them out, we are assuming that we are storing a duplicate URL in a separate list and then finally listing them out.

**Modification (Optimal Solution)**

Single Machine

What if the space is not available for the storage (i.e., average 100 characters each url and each character 4 bytes, So totally 4 TB). Is somehow we could have that much space then we separate the complexity by two pass algorithm. In first pass we separate the URL and split it in the capacity of 1GB each 4000 chunks and we can group similar hashed URL together in a file. In the second pass we push them into hash table and check for duplicates file by file.

Multiple Machine

If many machines can be leveraged then split the chunks and keep in separate machine and process manually and find out the duplicate.