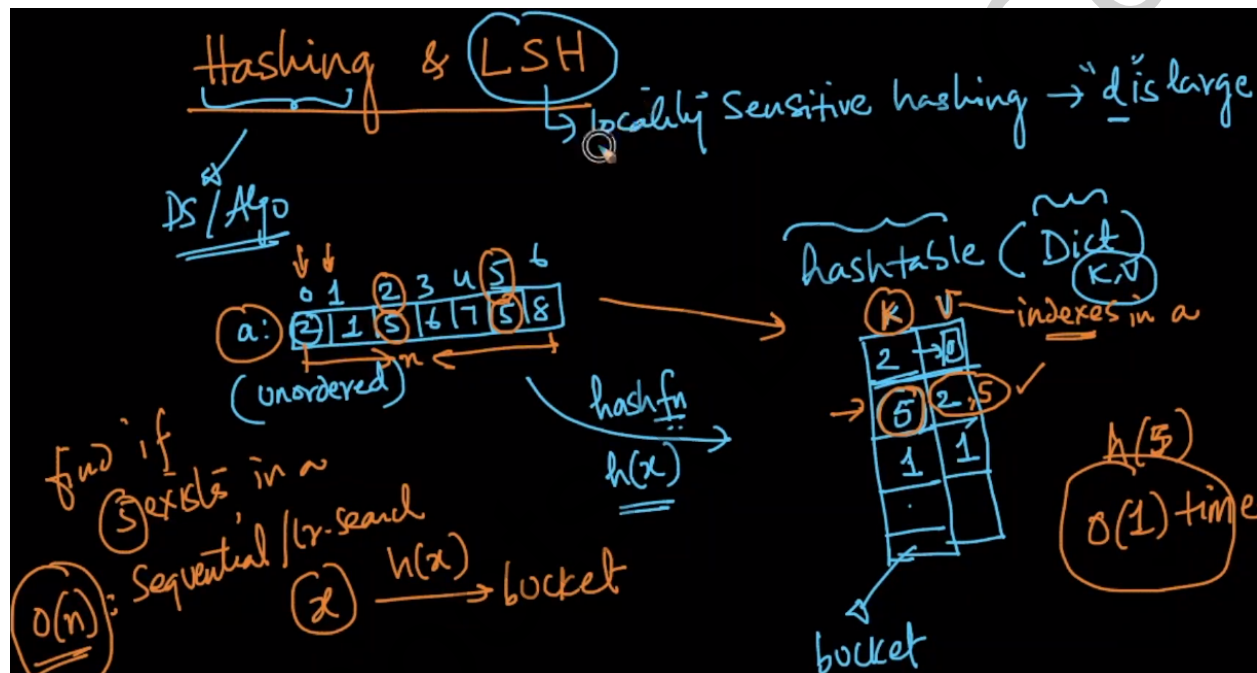


## 29.26 Hashing vs LSH(Locality Sensitive Hashing)

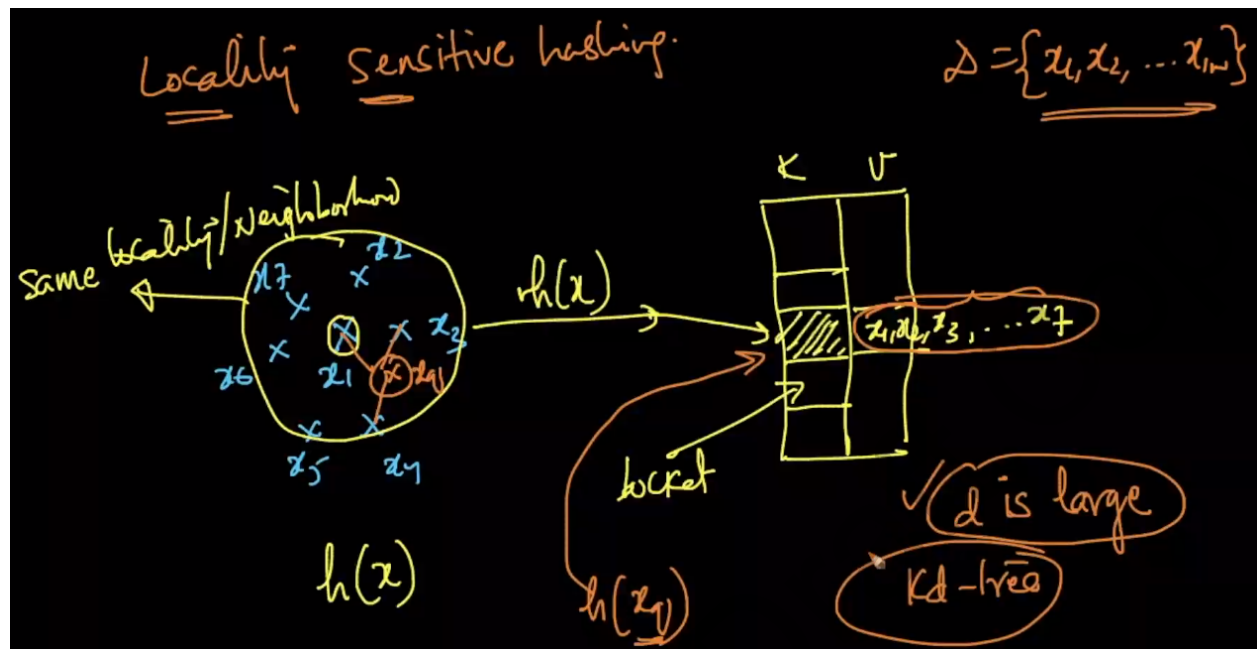
Locality Sensitive Hashing is a powerful technique when compared to KD-Tree and also works well when the dimensionality 'd' is large.

Before we look into LSH, we shall learn about Hashing. In general data structures, we have a data structure called Hashtable. The Hash function is the core concept on which a Hashtable works. One Hashtable will have only one hash function. There are no chances for one hash table to have multiple hash functions at a time. In python, Dictionary is the same as a hashtable.



The data in a hash table is stored in key-value pair format. If a value is given along with a key, then we can directly store that key-value pair in the hashtable. If a value is given without a key, then that value is passed through the hash function. The output of the hash function for a given point is the key, and each key is associated with a bucket in the hashtable. After we get the 'key' for a given value from the output of the hash function, then that particular value is stored in the bucket associated with the obtained key. So whenever we want to retrieve the value from the bucket, if we already know the 'key', then we can directly access it. Otherwise, we again have to pass the same value through the same hash function, we get the same key and then we can access it with that key. All the keys in a hashtable are unique, there is no duplication of keys.

Let us now look at the functioning of Locality Sensitive Hashing. Let  $h(x)$  be the hash function that is generated.



All the points here are in the neighborhood. For any given point ' $x_i$ ', in such a way that all the points that are nearby go to the same bucket in the hashtable. For the point ' $x_i$ ', first it has to go through the hash function, and a hash value is generated. This hash value will be the 'key' and ' $x_i$ ' will be the value. These two are going to be stored in the key-value pair format.

So here, all the points that lie in a neighborhood are stored in the cell associated with the same bucket. So we can easily pull out the ' $n$ ' nearest neighbors using LSH, as all these values are stored in the same cell.