

# Implementation Spec for Database

---

## Team Name - Ford Prefect

---

- Team Members

1. Tirth Upadhyaya(2018101069)
2. Naman Baheti (2018101099)
3. Hetav PAnchani(2018101016)
4. Divyank shah (2018101037)

## Implementation Details

---

- We looked at Some of the possible database structures in detail. But We implemented 2 level hashing with Red Black Trees Becoz of these Reasons
1. Tries- Is time efficient on all operations. We need to store additional information about the number of keys starting with each letter in all the nodes. The only concern is high memory usage. If at some later time we can optimize on memory we may try out this data structure.
  2. Fst- Current implementations of FST are not good for alpha-numeric keys. Also does not support changing the key value pair at a later time.
  3. Skip list: We used Red Black Trees becoz of optimized time complexity.

## Details of current implementations

### First Implementation :

- We first used two level hashing. The keys are first hashed by their first character which helps in memory and time optimization. So we have different hash tables for keys of different length, keys are further hashed based on their Length. Now each row of this hash table has a corresponding red black tree with the further data stored in it.
  - But we are comparing using lexicographic order of strcmp so our two level hashing will fail. So we need to change our Two level Hashing
  - We are using a two level hashing. The keys are first hashed by their first character which helps in memory and time optimization. So we have different hash tables for keys of different first letters, keys are further hashed based on their Second character. Now each row of this hash table has a corresponding red black tree with the further data stored in it.
  - We implemented Red Black Tree using algorithm given on this website <https://www.programiz.com/deletion-from-a-red-black-tree>
  - Our time Complexity for various operations are
1. Put  $O(\log(n))$

2. get  $O(\log(n))$
3. del  $O(\log(n))$
4. deln  $O(\log(n))$
5. getn  $O(\log(n))$

### **Final Implementation**

So we finally used Red Black tree only as our base data structure in our hash table

### **Memory Allocation :**

- We are currently using malloc for memory allocation when we need.
- We are not using bulk allocator as it would be troublesome in case of parallel threads

### **Optimizations :**

- We didn't use vectorized functions becoz they are taking more time than normal functions.
- We also have done two Level hashing which should save us  $O(\log(52)*\log(52)) * 10000000$  iterations during insertion