Properties of Tries

Theorem: The linked structure of a trie is independent of the key insertion/deletion order: there is a unique trie for any given set of keys.

Worst-case time bound for search and insert

Theorem on Search: The number of array access when searching in a trie or inserting a key into a trie is at most 1 plus the length of the key.

Theorem on Search miss: Most of the time, you can say the search is a miss after very few examinations. ~logN with base R. N is the number of keys.

From a practical standpoint, the most important implication of this proposition is that search miss does not depend on the key length. For example, it says that unsuccessful search in a trie built with 1 million random keys will require examining only three or four nodes, whether the keys are 7-digit license plates or 20-digit account numbers.

Theorem on space. The number of links in a trie is between RN and RNw, where w is the average key length.

application	typical key	average length w	alphabet size R	links in trie built from 1 million keys
CA license plates	4PGC938	7	256	256 million
account numbers	02400019992993299111	20	256	4 billion
			10	256 million
URLs	www.cs.princeton.edu	28	256	4 billion
text processing	seashells	11	256	256 million
proteins in genomic data	ACTGACTG	8	256	256 million
			4	4 million

Space requirements for typical tries

The bottom line is this: do not try to use this trie implementation for large numbers of long keys taken from large alphabets. Otherwise, if you can afford the space, trie performance is difficult to beat.