CS 310 – Advanced Data Structures and Algorithms

Searching

June 14, 2018
Searching

- **Linear Search**
  a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned.

- **Hash Table**
  \(O(1)\) for lookup

- **Binary Search**
  Binary search finds the position of a specified value within a sorted array.
  Every iteration eliminates half of the remaining possibilities. This makes binary searches very efficient. 
  Worst case performance: \(O(\log n)\)
public int binarySearch(int[] A, int key) {
    int start = 0;
    int end = A.length - 1;
    while (start <= end) {
        int mid = start + (end - start)/2;
        if (key == A[mid]) {
            return mid;
        }
        if (key < A[mid]) {
            end = mid - 1;
        } else {
            start = mid + 1;
        }
    }
    return -1;
}
The word trie comes from retrieval. Also known as Prefix/Radix/Digital Tree

Trie is relating to Most-significant-first radix sort.

Using trie, search complexities can be brought to optimal limit (key length)

The root represents an empty string

Every node of trie consists of multiple branches

Each branch represents a possible character of keys.
Tries

From wikipedia
Tries

Why use tries if hash tables can do the same?

- Hash tables can only find in a dictionary words that match exactly with the single word that we are finding.
- The trie allow us to find words that have a single character different, a prefix in common, a character missing.