One downside of the 27-Way Tree data structure is the huge memory cost of storing 27 links per node, because most of the links are unused. The other downside is that it takes O(L) to add and search a word, while L is the length of given string. I choose to realize a resizing chaining hash map to simulate dictionary, more memory efficient. Assuming entries are even spread in hash map, time complexity of dict\_addword() is O(1), while key copy for resizing takes O(n).

Hash map cannot realize autocomplete method efficiently, as 27-Way Tree and other data structures similar to trie can be the best choice for storing and searching for prefix. However, considering the great memory efficiency in hash map, this extension version can be a foundation of realizing a different 27-Way Tree with 'Down' pointers point to hash maps instead of arrays.