

Data Structure and Advanced Programming

Homework #10

Due: 2021/5/25 08:00am (CST)

NOTE: Please upload your answers in either English or Chinese as a PDF to NTU COOL before the due date and time.

1. (30%) When you use a word processor's spell checker, it compares the words in your document with words in a dictionary. You can add new words to the dictionary as necessary. Thus, this dictionary needs frequent retrievals and occasional insertions. Which implementation of the ADT dictionary would be most efficient as a spell checker's dictionary? Please consider a sorted array-based implementation, a sorted linked-based implementation, an implementation that uses a binary search tree, and an implementation that uses hashing, respectively, and write your answer with the reason.
2. (20%) Demonstrate the insertion of the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \bmod 9$.
3. (20%) Consider a hash table of size $m = 1000$ and a corresponding hash function $h(k) = \lfloor m(kA \bmod 1) \rfloor$ for $A = (5^{1/2} - 1) / 2$. Compute the locations to which the keys 61, 62, 63, 64, and 65 are mapped.
4. (30%) Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length $m = 11$ using open addressing with the primary hash function $h_1(k) = k \bmod m$. Illustrate each step of the results of inserting these keys one by one using linear probing with $h(k, i) = (h_1(k) + i) \bmod m$, quadratic probing with $h(k, i) = (h_1(k) + i^2) \bmod m$ for the i -th probe position, and double hashing with $h_2(k) = 1 + (k \bmod (m - 1))$, respectively.