

french.txt



(Clearer)

french.txt

In task 7, we were tasked to test how long it takes for the program to read a text file and place it in the hash table. Due to the variation of base and table sizes, the possibility of collision might occur hence, collision in task 7 was handled using separate chaining with a binary search tree.. With a time limit of 2 minutes (120 seconds), different bases with different table size was tested against each other and was compared with each other.

According to the data of the graph, using separate chaining, when the base of the hash table is too small compared to the table size, the time taken for the program to insert all words into the hash table takes longer. It occurs for all text file, english\_small.txt, english\_large.txt and french.txt. When the base is too small, the difference between the hash value will be too small, hence there will be a higher chance for collision to occur. However, it is faster than linear probing and quadratic because in separate chaining, probing is not required, when a collision occurs, the value and key is inserted into the left or right child of the tree, depending on the hash value.

When the value of base of the hash table and the table size is too close to each other, the time taken for the program to insert all the words into the hash table also takes longer. When the value of the base and the table size are too similar, the hash value obtained may be too big, hence there will be a higher chance for collision to occur. . However, it is faster than linear probing and quadratic because in separate chaining, probing is not required, when a collision occurs, the value and key is inserted into the left or right child of the tree, depending on the hash value.

According to the graph, collision occurs less frequently when there is a big value difference and a small value difference between the base and table size. This is because collision is only calculated when the program enters the set item function again, not when it repeats the for loop. As collision is handled through separate chaining, less time is required to handle the collision as when a collision occurs, the value and key is inserted into the left or right child of the tree, depending on the hash value. Therefore it is faster than linear and quadratic probing.

According to the graph, the length of the probe is longer when there is a big value difference and a small value difference between the base and table size. The total probe in this separate chaining is calculated through counting the length of the binary tree. Hence, if collision occurs frequently, the total probe will be high.

The length of the probe is shorter when the value difference between the base and table size is not too small and not too big because the frequency of collision is lower hence, the length of the binary tree is shorter, total probe is shorter

According to the graph, through separate chaining, the max of the probe is bigger when there is a big value difference and a small value difference between the base and table size. The total probe in this separate chaining is calculated through counting the length of the binary tree. Hence, if collision occurs frequently, the total probe will be high. Hence, the value of the max probe will also be higher.

The length of the probe is shorter when the value difference between the base and table size is not too small and not too big because the total probe in this separate chaining is calculated through counting the length of the binary tree. Hence, if collision occurs less frequently, the total probe will be low. Hence, the value of the max probe will also be lower.