Robert Alvarez

Advance Algorithms, Assignment 4

1. Compile the given code found in archive assignment4.tgz, containing the linked list implementations, the hash tables, and the dictionary driver code. Run the code on file sp-en-dictionary.txt. Observe the number of collisions. Modify the code repeatedly to lower the array size used for the hash table. Run the code each time, observing the number of collisions. Draw a graph giving the number of collisions (as a percentage of the number of entries) in the function of the hash table size. Describe the graph and its overall behavior.
   1. Analysis of the number of collisions.
      1. A graph with blue dots

         Description automatically generated
      2. By looking at the number of maximum collisions in a hash table entry for different hash table sizes, we can observe that as the hash table size increase, the # of collision decreases to no less than 6. Testing was done for sizes above 100,000; the minimum never exceeded 6. This is due to the key (Spanish word) distribution.
   2. Analysis of unused entries.
      1. A graph with blue dots

         Description automatically generatedA graph with a line

         Description automatically generated
      2. This graph shows the memory usage inefficiency for large hash table sizes. So, even if graph (a) keeps decreasing the number of collisions per entry use (to a minimum of 6), the hash table starts to have more and more unused entries concerning the total number of entries.
   3. Analysis of actual vs expected # of collisions.
      1. A graph with a line

         Description automatically generatedA graph with blue lines

         Description automatically generated
      2. This graph shows that the key distribution behaves uniformly for relatively small hash table sizes because the ratio between the actual (worst case) and the expected # of collisions is relatively close to 1.0 (perfect hashing in uniform key distribution).
   4. Analysis of % of collisions per entry usage.
      1. A graph with blue dots

         Description automatically generatedA graph with blue dots

         Description automatically generated
      2. By analyzing the percentage of collisions per entry, we can derive a similar reasoning as in analysis (b) because the % of collisions per entry (worst case) is close to 100/hashtable\_size.
2. Compile the given code for the size of an array with 16 entries. Run the code on the mini dictionary sp-en-mini.txt. With the use of gdb, draw the graph of pointers of objects created in memory for this dictionary.

