CS 32 Homework 4

2 - URL did not overload a != operator. When insert is called, it calls the find function on the URL. In the find function, it tries to compare one URL with another, however, the != operator was not defined for a URL so the compiler does not know how to compare two URLs.

3 b – If we had to implement this function recursively with only one parameter, we can’t solve the problem because we have no way of knowing what the previous string which represents the parent nodes above it.

4 a - The big-O of the algorithm is O(N3). This is because there is a loop within a loop within a loop, each of which goes through N iterations. The result of this would be N\*N\*N which is O(N3).

4 b - The big-O of the algorithm is O(N3). This is because the first loop goes through N iterations, and in the WORST CASE SCENARIO, the second loop goes through N iterations, and the third loop goes through N iterations. The result would be N\*N\*N so the big-O of the algorithm would be O(N3)

5 a – The big-O of the algorithm is O(N3). This is because the first loop goes through N iterations(the unique size of the Multiset). Within the for loop, the function get is called, which goes through N/2 iterations in the worst case and it also calls contains, which goes through N iterations. Contains also calls the insert function which goes through N iterations. Because of this, the equation would be N3 + N2/2 but since constants and lower terms are ignored, the big O of the algorithm is O(N3). Swap was not considered because it is a constant time function.

5 b – The big-O of the algorithm is O(N logN). This is because the first for loop goes through N iterations. So far our equation is N. The second for loop, which is not inside the first for loop, goes through N iterations. So far our equation would be 2N. We simplify this to N by ignoring the constants. When we call sort, we go through NlogN iterations. So far our equation is N logN + N. We simplify this to N log N by cutting out the lower terms. The next loop goes through N iterations. Everything in the next loop is a constant time function so our equation would be N log N + N, simplified to N log N. In the if statement, there is a do while loop that goes through at most N iterations. Because this term is not greater than N, our big O is O(N3).