HW 4

2. The insert() function calls the find() function to search for the URL that has been passed in as a parameter. Find() tried to loop through the Multiset checking if p->m\_value != value. Since value is a URL object and there is no != operator defined to compare URLS, the compiler runs into an error.

3b. Without two parameters, there would be no way to record the previous path that should be printed out before printing the name of the current submenu. It would only be possible to print out one name at a time rather than the whole preceding hierarchy for each name.

4a. O(N^3) because there are three nested for loops on a 2D array. The int i iterates through 0 to n-1, and for each i the int j visits 0 to n-1. The third loop with int k is visited only for n-1 of the j values in the middle loop (since if i == j, the third loop is skipped). The algorithm is approximately n\*(n-1)\*n , which is approximately O(N^3).

4b. O(N^3). Even though this algorithm is faster than part a, it still goes through three loops. It’s only faster because the third loop doesn’t need to visit all n terms, but ignoring any constants, the overall time complexity is still O(N^3).

5a. O(N^2) because the function itself loops through m1 once, but within this outer loop, the function also calls contains() which loops through m2. The other operations in the function can be ignored because they are all either of a smaller order or constant time.

5b. O(NlogN) because there are no nested loops in this implementation. The individual loops are O(N), with an additional sort function of O(NlogN). 4N+NlogN averages out to NlogN. This is better in terms of time complexity compared to part a, which used nested loops.