Project 4 Report

Clayton Lewis

1. I used two hash table to store the information in synsets.txt. In the first hash table, the keys are strings and the values are ArrayLists of integers. I chose this so that I could map each noun to a list of all synsets of which it is a part. In the second hash table, the keys are integers and the values are strings. I chose this so that I could map the id of each synset to the string representing the nouns comprising that synset. I also chose to use hash tables because they have the distinct advantage of accessing elements in logarithmic time (constant time for the synsets hash table because each id can map to only one synset).
2. I used a directed graph (the given Digraph class) to store the information in hypernyms.txt. I made this choice because hypernyms can be represented as edges in a directed graph. In the given implementation, a vertex can be accessed or an edge added in constant time.
3. My SAP algorithm first creates a BreadthFirstDirectedPath using the directed graph and the given vertex v, then does the same for the given vertex w. That is, a breadth first search is performed separately for each vertex, Then every vertex in the graph is checked. If it was marked in the breadth first searches for both vertices, it is added to a stack. The stack will contain all common ancestors of the two vertices. Then every ancestor in the stack is checked to find the ancestor of shortest path length. The common ancestor with the shortest path is returned. If none was found, -1 is returned.
4. The worst case for the SAP algorithm occurs when every vertex in the directed graph is connected and is a common ancestor of the given vertices. In this case, it takes $E = V^2$ compares to perform each of the breadth-first searches, V compares to check every vertex in the graph, and V compares to check every common ancestor and find the ancestor with the shortest path, where V is the number of vertices and E is the number of edges in the directed graph. Thus the worst case running time is $\sim E = V^2$.
5. The best case for the SAP algorithm occurs when the given vertices have no ancestors at all. In this case, it takes ~1 compare to perform each of the breadth-first searches, V compares to check every vertex in the graph, and 0 compares to check every common ancestor, where V is the number of vertices in the directed graph. Thus the best case running time is ~V.