Justin Moy

I used the same data structure because I already had the code for it made already. I also used it because it seemed like the only viable one to use for the Floyd-Warshall(FW) Algorithm. The average and worst case runtimes are the same for my structure in Bellman-Ford(BF). I used an adjacency matrix, so every index in the matrix was always checked so every edge could be checked V-1 times. SO it checks V * V items V-1 time giving the run time of O(V^3) . Using an adjacency list would change the average runtime, so only edges get checked V-1 times. This improves runtime to O(E * V). Worst case is the same if there is an edge between every vertex then the matrix will have a size similar if not equal to an adjacency matrix, which will give it the same runtime in a worst-case. Worst and average case runtime for Floyd-Warshall is O(V^3) because it uses 3 for loops, each run V times, and only one action is performed and that is checking to see if $A[i,k] + A[k,j] < A[i,j]$ and assigning it to $A[i,j]$ if it is. Neither time can be improved because of how the algorithm is structured. An adjacency matrix must be used, and it is nearly, if not is, impossible to do with a list. This is the optimized runtime. Space complexity is O(V^2) for BF for both average and worst-case because that is the size of the matrix. A list would improve the average space complexity to O(E) since there is only a spot created in the list if the edge exists, but for worst-case becomes O(V^2) if every vertex is adjacent to each other. For FW space complexity, average and worst is O(V^2) because it is using a matrix of size V^2. This cannot be improved because the algorithm would not work with a list. Since the list only holds items if the edge exists then nodes that don't directly connect won't exist in the list whereas with a matrix they get an edge cost of infinity. If the list did do what the matrix did then its size would be equal to the matrix. Reusing my data structure made everything work well and coding easier, I encountered no problems at all.