Dylan Bickram

Amanpreet Kapoor

COP3530

Section: 1085

Report

InsertEdge Function:

The insertEdge function implemented in my PageRank project has a time complexity of ~O(outdegree|V|) and a space complexity of O(|V| + |E|) in the worst case. The reasoning behind insertEdge's time complexity is that every vertex in the set V is assigned an edge directed towards another vertex through pushing back an directed vertex in a vector associated with it. That act of pushing back occurs |V| as by the end every vertex relates to an edge. Thus, the function insertEdge has a time complexity of ~O(outdegree|V|). In terms of space complexity, the reasoning behind O(|V| + |E|) as the functions uses space to create a 'to' and 'from' vertex every time the function is called as well as the edges to needed to connect vertices together. The vertices are created by create Vertex objects and as well as an increment on the variable that holds the total number of vertices in the adjacency list. The edges are formed by pushing back space onto a vector for each individual node; thus, leading to |V| + |E|;

PageRank Function:

The PageRank function has a time complexity of O((|V||E|)*p + |V|) and a space complexity of O(|V| + |E|). When discussing time complexity, the p in the big-O notation represents the parsed power iteration input as the page rank starts with a for loop that iterates from 1 to p-1 that reduces to p in big O notation. Then the function iterates through every webpage in the adjacency list, which represents |V|. There is another function that also iterates through every webpage in order to create the iteration matrix, which causes the time complexity to change to |2V|. However, |2V| is reduced to |V|. Within the previous for loop there is a function that iterates through the edges of a given of a vertex, and this will occur for every vertex in the adjacency list, leading to my conclusion of |E|. There is also another loop that inside the power iteration loop that also spanned to all the vertices in the adjacency matrix which would take |V| time, leading to an overall time of O((|V||E| + |V|)*p); however, the plus |V| can be dropped to have a time complexity of O((|V||E|)*p). The PageRank function has another function that prints out he webpage and the page rank; the printPageRank() function. The printPageRank() function iterates through the entire adjacency list touching every vertex in the list, thus the function has a time complexity of |V|. Therefore, the overall time complexity of the PageRank function is O((|V||E|)*p + |V|). When referring to the space complexity that for the PageRank function, there are two main factors. One, a double vector was created in the function that holds the iteration matrix, which has a maximum size of |V| for all the vertices in the list. Two the helper function printPageRank function creates a sorted map of the previous unsorted adjacency list, thus it holds a max of |V| + |E| space. With that in mind if we combine |V| and |V|+|E| the big-O notation of PageRank is O(|V| + |E|) as the two that would have been in front of the |V| was dropped.