Ali-Sarosh Bangash

Faiz Alam

Sumit Jadhav

11/20/2022

Data Structures and Algorithms

Dijkstra’s Algorithm

Dijkstra’s algorithm is an algorithm used to find the shortest path between two nodes represented by the graph abstract data type. It was developed by Dutch computer scientist Edsger W. Dijkstra in 1956. The procedure for Dijkstra’s algorithm follows like this: let the node we start from be called the “starting node” and the node that we are trying to search for be the “end node”. Now let the distance between the starting node and the end node be a variable, called “shortest path” or shortPath. We mark all the nodes we haven’t visited as “unvisited” and set their distance to infinite, while we set our initial/starting node to a distance value of 0. As we run our algorithm, our shortPath will be continuously updating as we go through our graph and pick out all the nodes we visit and calculating their distances with respect to the current node. In other words, we are performing a procedure of “relaxation” where we take the sum of the distance that we already have with our current node plus the distance of the edge connecting the current node to an adjacent node, and then seeing if that sum is smaller than the distance already at the adjacent node. For instance, if the distance at some current node A is 5 and the distance of adjacent node B is 10, and the distance of the edge connecting these two nodes is 3, then 5 + 3 = 8, and 8 < 10, so we update adjacent node B to the value of 8. Once we are done considering all of the adjacent nodes of the current node, we mark the node as “visited” and remove the node from the set of all “unvisited” nodes that we had initially. This is done so that the ”visited” node will never be checked again, since we’ve already determined the smallest distance for the current node by the process of “relaxation”. If the final/destination node is marked “visited”, then we stop the algorithm. If not, we set the unvisited node as “current” and go back to our previous process of “relaxation”. The time complexity for Dijkstra’s algorithm is O(|E| \* log|V|), where |V| represents the total number of vertices of our graph, and |E| is the total number of edges within our graph.

Dijkstra’s algorithm has many useful applications in the real world. For instance, it is extremely useful for applications in the airline industry, where airliners try to determine the shortest path from one airport to the other. It is also applied to digital mapping services like Google or Apple Maps, as it can be used to determine the quickest path from one city/place to another city/place.

In conclusion, we have illustrated Dijkstra’s algorithm through a detailed step-by-step description of the algorithm at every point. We’ve also mentioned the time complexity for Dijkstra’s algorithm. We’ve also detailed how Dijkstra’s Algorithm is used in the real world, through services like Google and Apple Maps, and through how airliners try to reach other airliners in the shortest way possible.

**References**

Baeldung. (2022, November 9). Understanding time complexity calculation for Dijkstra algorithm. Baeldung on Computer Science. Retrieved November 20, 2022, from <https://www.baeldung.com/cs/dijkstra-time-complexity>.

Dijsktra's algorithm. GeeksforGeeks. (2022, August 31). Retrieved November 20, 2022, from https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/.

Wikimedia Foundation. (2022, November 1). Dijkstra's algorithm. Wikipedia. Retrieved November 20, 2022, from <https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm>.

Bari, A. (n.d.). 3.6 dijkstra algorithm - single source shortest path - youtube. Retrieved November 21, 2022, from <https://www.youtube.com/watch?v=XB4MIexjvY0>.