Analysis of Dijkstra's Algorithm

Hu Tianrun, Ian Chan Jiajun, Kelvin Pang Junwei

September 18, 2021

The Dijkstra's algorithm maintains the min-priority queue Q by calling three priority-queue operations: ADD, POP, and $MAINTAIN$. The algorithm calls both ADD and POP once per vertex. Because each vertex $u \in V$ is added to set S exactly once, each edge in the graph is examined in the loop exactly once during the course of the algorithm. For the adjacency list graph, since the total number of edges in all the adjacency lists is $|E|$, this loop iterates a total of $|E|$ times, and thus the algorithm calls $MAINTAIN$ at most $|E|$ times overall.

The running time of Dijkstra's algorithm depends on how we implement the min-priority queue and graph. In this project, I implement graph by adjacency list and matrix, and combine them with heap priority queue and array queue respectively.

Consider first the case in which we using array as queue, maintaining the min-priority queue by taking advantage of the vertices being numbered 1 to $|V|$. We simply store distance of each v in an array. Each ADD operation takes $O(1)$ time and POP operation takes $O(|V|)$ times. For this case, there is no need to have a $MAINTAIN$ function, in another word, the $MAINTAIN$ function is combined with POP function. The time complexity for matrix with array queue is $O(|V|^2 + |V|^2) = O(|V|^2)$. The time complexity for adjacency list with array queue is $O(|V|^2 + |E|) = O(|V|^2)$.

We can improve the algorithm by implementing the min-priority queue with a min-heap. Each time we add a vertex into queue, we need to call $MAINTAIN$ function to maintain the queue. And each time we pop the vertex, we need to call $HEAPIFY$ function to keep heap property. Both $MAINTAIN$ and $HEAPIFY$ take $O(\lg |V|)$ time. What's more, every time we change the distance of a vertex, we need to call $MINTAIN$ function to maintain the heap that at most have $|E|$ times. For the adjacency list graph, the time complexity is $O((2|V|+|E|) \lg |V|)$, and for the matrix list graph, the time is $O((2|V|+|E| \lg |V|+|E|) = O((2|V|+|E|) \lg |V|)$

For conclusion, we can see that the asymptotic time for matrix graph and adjacency are always the same and using min-heap is asymptotically better than array priority queue. However they still have difference performances in different cases by some constant times. This part will be shown in our data analysis.