public, public transit GuiFengYe

(a) Dijkstra’s algorithm to find the shortest path from a single source vertex to all other vertices in the given graph.

consider the starting time as root , the length(e) which travel from u to v as weight. (but also affected by waiting time) . I will try to find the minimum weight from root S to Final station S.

(1) Create a set sptSet (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

3) While sptSet doesn’t include all vertices

….a) Pick a vertex u which is not there in sptSet and has minimum distance value.

….b) Include u to sptSet.

….c) Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.)

(b) Time complexity should be O(V^2)

(c)Dijkstra’s algorithm

(d) If the input graph is represented using adjacency list, with the help of binary heap, complexity can be reduced.

(e) O(E log V)

Min Heap is used as a priority queue to get the minimum distance vertex from set of not yet included vertices.

1) Create a Min Heap of size V where V is the number of vertices in the given graph. Every node of min heap contains vertex number and distance value of the vertex.

2) Initialize Min Heap with source vertex as root (the distance value assigned to source vertex is 0). The distance value assigned to all other vertices is INF (infinite).

3) While Min Heap is not empty, do following

…..a) Extract the vertex with minimum distance value node from Min Heap. Let the extracted vertex be u.

…..b) For every adjacent vertex v of u, check if v is in Min Heap. If v is in Min Heap and distance value is more than weight of u-v plus distance value of u, then update the distance value of v.