Dijkstra’s Algorithm

Solve single-source shortest path problem.

The algorithm finds the shortest path from source vertex v to all other vertices in a graph.



Initialize:

Q structure (not a regular q, just a container): A, B, C, D, E

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (SOURCE) | B | C | D | E |
| 0 | INFINITE | INFINITE | INFINITE | INFINITE |

S = { },

Dist[y] stores the currently calculated shortest length(distance) from source vertex (A in this case) to vertex y.

Iteration 1:

* Dequeue (remove and return) from Q the item with the smallest distance value.
* (In this case, A is removed from Q.)
* Then Add it (in this case A) into set S. (After we add vertex A into S, that means that we have already found the shortest path to A). S= {A}
* Check each of its (in this case A’s) reachable neighbors (In this case B and C), update their dist value. (Because the shortest path to A is found, A’s Neighbor might be brought closer to the source vertex. In other words, a new path may be found through vertex A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (SOURCE) | B | C | D | E |
| REMOVED | 10 {Path(A) + AB} | 3 {Path(A) + AC} | INFINITE | INFINITE |

S = {A}

Iteration 2 - same as iteration 1:

* Dequeue (remove and return) from Q the item with the smallest distance value.
* (In this case, C is removed)
* Then Add it (in this case C) into set S. S={A, C}
* Check each of its (C’s) reachable neighbors (B, D, E), update their distance value.
  + We grab the distance to our current node (whatever we just added into s) and add that to the neighbors’ weights. If the resultant sum is lower than what is currently stored in the Q for that given vertex then the Q is updated with the lower value.
    - We do this step in iteration one as well, it’s just here instead of above for ease of explanation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (SOURCE)- REMOVED | B | C - REMOVED | D | E |
| 0 | 7 {path(C) + CB} | 3 {Path(A) + AC} | 11 {path(C) + CD} | 5, {path(C) + CE} |

S = {A, C}

Wash rinse repeat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (SOURCE)- REMOVED | B | C - REMOVED | D | E - REMOVED |
| 0 | 7 {path(C) + CB} | 3 {Path(A) + AC} | 11 {path(C) + CD} | 5, {path(C) + CE} |

S = {A, C, E}