Shortest Path Problem

James Zhang 400076580

Nov/19/2019

Shortest path problem is one of the most famous problems in graph theory. This problem involves a special type of graph called weighted directed graph. A weighted graph is the graph that every edge has its own cost. A directed graph is a special type of graph such that each edge has a direction, when moving along the edges it must follow the direction. The key concept of this problem is looking for the shortest directed path from starting vertex s to destination vertex t, such that the cost of the used edges is minimized.

One of the best algorithms that can solve this problem is call Dijkstra's algorithm. It was designed by Edsger W. Dijkstra in 1956. It is famous because comparing to all the other algorithms existed, it has the lowest average case complexity, which means Dijkstra's algorithm is the fastest method to solve the shortest path problem on a weighted direct graph. It saves significant amount of cost when processing large size of data. This algorithm is easier to understand for the people with a little computer science knowledge. It can be described in English as:

1. Define two sets, visited and unvisited. visited is empty at the beginning. All the vertices are in unvisited. All the vertices contain an information, that is the distance to the starting vertex. All the edges store an information that is either marked or unmarked. All the edges are unmarked initially. Set the distance to infinity for all vertices, except for starting vertex it is 0.

2. Look for the smallest vertex in unvisited set. Update the new distance to the adjacent vertices if the new distance is smaller. Mark the edge that forms a smaller distance. Move this smallest vertex from unvisited to visited.

3. Keep repeating step 2. until the destination vertex is in visited set. Then a shortest path is formed by the marked edges.

This is super hard to understand by just looking the description. A good example is then needed for a better understanding. In $Figure 1$, a is the starting vertex, while f is the destination. All edges have different weights. It is nicely set up. Dijkstra's algorithm can then be used to find the shortest path.

The vertex with the smallest distance is the starting point a. For all the vertices that are adjacent to a, calculate their distance, replace the old distance with the new distance if the new distance is less. In this case, all the adjacent vertices b, c, d have old distance of ∞ so replace them all. For the edges that direct to the vertices with changed distance, mark them to red. Move vertex a to visited set. This operation is shown in $Figure 2$.

Move on to the next vertex, look for the vertex with the lowest distance in the unvisited set. It is b with distance of 5. b has two adjacent vertices d and e. Calculate the new distance of d: $5 + 16 =$

21, which is greater than the old one which is 19. Therefore, nothing needs to be changed for d since the rule is only replace the old distance to a smaller one. Move to the other vertex which is f: $5 + 6 = 11$. Replace infinity with 11 since $11 < \infty$. Also mark the edge $b{-}e$ to red since the distance is updated. At the last, move b from the unvisited set to the visited set. This operation is shown in $Figure 3$.

Keep repeating last step: look for the vertex with the lowest distance in unvisited set. It's c with distance of 10. Calculate the adjacent vertex d, the new distance of d is $10 + 7 = 17$. Replace the old distance with 17 because $17 < 19$. Then change the edge $c{-}d$ to red, and change a-d back to black because $c - d$ is now the edge that gives d a smaller distance. At the end, move c from the unvisited set to the visited set. This operation is shown in $Figure 4$.

Now just keep doing the same operation for vertices e. Replace distance of f with 22. Mark $e - f$ to red. Move e from unvisited to visited. This operation is shown in $Figure 5$.

Then visit d. Nothing needs to be changed since $17 + 16 > 22$. Move it to visited. The destination vertex f is then visited. Nothing needs to be changed. Just move it to the visited set. Then a shortest path is formed by the red edges, which is $a - b - e - f$, with distance of 22. This operation is shown in $Figure 6$.

The algorithms to solving the shortest path problem has mickle of applications, such as concurrency transportation, map guiding system, long-distance telephone calls routine and linear programming. Data scientists and mathematicians really paid big efforts to make our lives better by designing those complicated algorithms. It is also interesting for students to study further.

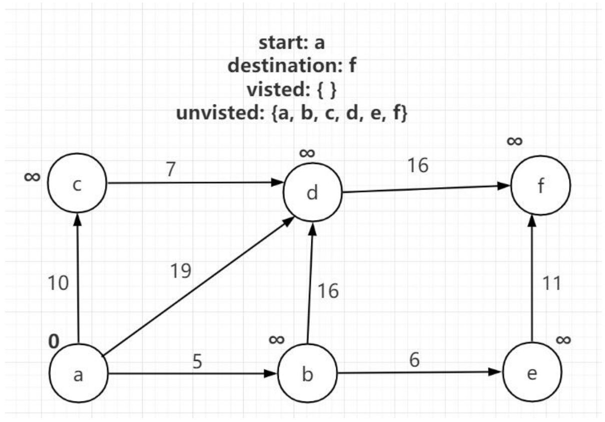

Figure 1: Algorithm Example

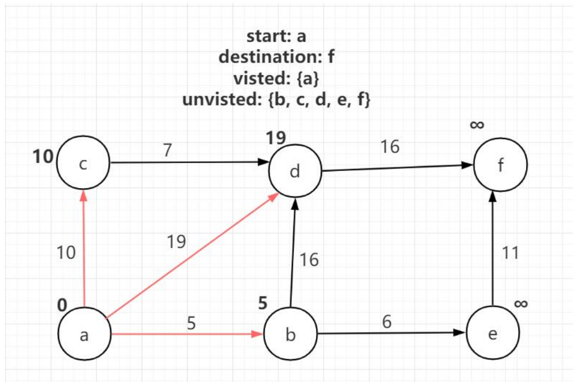

Figure 2: Visting a

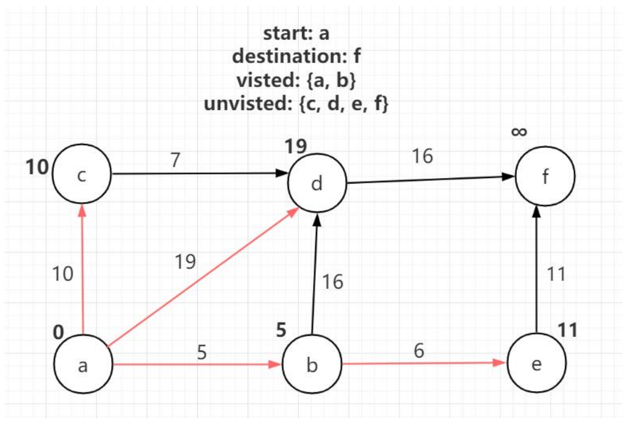

Figure 3: Visiting b

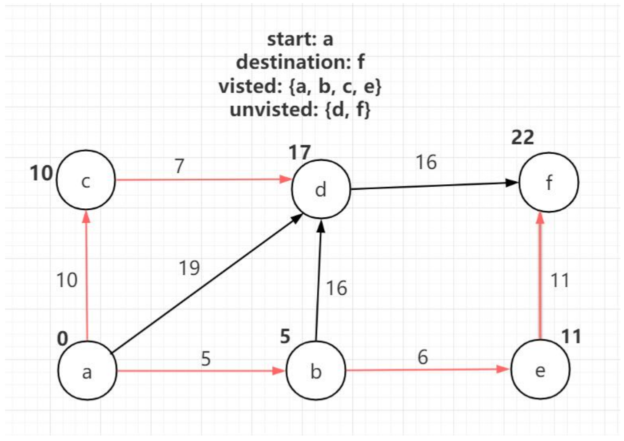

Figure 5: Visting c

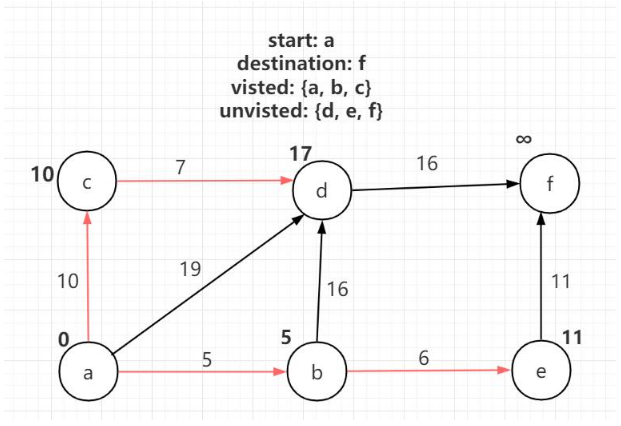

Figure 4: Visting c

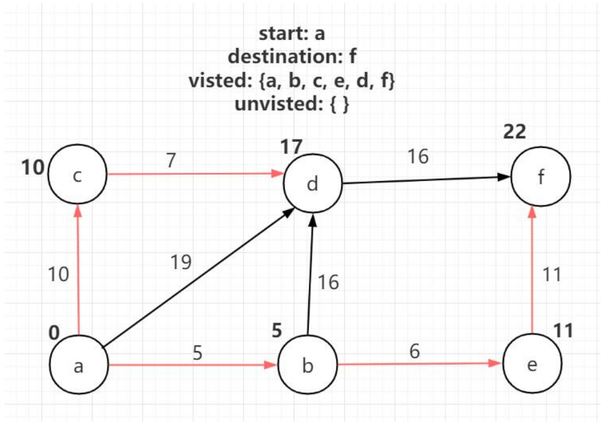

Figure 6: Visting d, f

References

[1] Javaid, Adeel. (2013). Understanding Dijkstra Algorithm. SSRN Electronic Journal. 10.2139/ssrn.2340905.

[2] Ford, L.R. and Fulkerson, D.R. (1958) Constructing maximal dynamic flows from static flows. Operations Research 6, 419–433.

[3] Winston, W.L. (2004) Operations Research Applications and Algorithms, Fourth Edition. Brooks/Cole, Belmont, CA.

[4] Dijkstra, E.W. (1959) A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, 269–271.