Dijkstra's Shortest Path Algorithm Made Simple

Daniel S. Wang

Some Definitions

1. *SolutionSet* - Set of vertices that have finalized shortest paths.
2. $SD[v]$ - Shortest Distance from source to v.
3. $Parent[v]$ - The parent of v on the shortest path from *source* to v.
4. Fog of War - Anything not in the SolutionSet
5. V - Set of vertices in the graph
6. *source* - The originating vertex

```
Dijkstra's Algorithm (V, source){
  SolutionSet.add(source)
  SD[source] = 0
  V.remove(source)

  /* Initialize */
  for (v in V) {
    SD[v] = weight(source, v) // Or INT_MAX if the edge doesn't exist
    Parent[v] = source
  }

  /* Main Algorithm */
  while(not all vertices are in the SolutionSet) {
    int minDist = INT_MAX
    Vertex v

    /* Select vertex in Fog of War that is shortest distance from an existing vertex */
    for (vertex in V) {
      /* Hint: Most SD[vertex] will be INT_MAX */
      // The vertices at the border of the Fog of War will have SD's less than INT_MAX
      if (!SolutionSet.contains(vertex) && SD[vertex] <= minDist) {
        minDist = SD[vertex]
        v = vertex
      }
    }

    SolutionSet.add(v)
    V.remove(v)
```

```
    /* Explore */
    /* new territory in the Fog of War */
    /* through v */

    /* by updating weights with new weights from v */
    for (unvisited in V) {
      int newWeight = SD[v] + weight(v, unvisited) // weight = INT_MAX if doesn't exist
      if (newWeight < SD[unvisited]) {
        SD = newWeight
        Parent[unvisited] = v
      }
    }
  }

  return SD, Parent
}
```