Data Structures
Djikstra Algorithm

Cain Susko

Queen's University
School of Computing

March 30, 2022

Finding the Shortest Path

it is useful to know the shortest path in an a graph is useful as it allows us to move as efficiently as possible in a graph. A graph will normally be wieghted if you are asked to find the shortest path, the objective is to find the path from a to b with the smallest sum of weights.
One can also be taked to find the shortest path between any 2 verticies in a graph.

Djikstra's Algorithm

This algorithm determines the shortest path in G from vertex s to **all other** verticies in the graph.

Input	G, s
Output	$dist$

where $dist$ is an array containing the optimal distances from s to every other vertex in G.

Data Structure

Djikstra uses 3 different data structures to carry out its operation.

1. A $dist$ array (min-heap) that tracks the **current best known cost** to get from s to every other vertex in the graph. For each vertex i in $dist$:

   (a) set the distance from s to be equal to 0

   (b) set the distance from all other verticies to ∞

2. An array called $done$ that holds boolean values denoting if a vertex has been fully prosessed. For each vertex i, set $done[i]$ to false.

3. An array called $parent$ that holds the parent for each vertex.

Algorithm

The Djikstra Algorithm is as follows:

```
while !done {
        u = the closest unprocessed vertex to s

        for v in notDoneVertx{
                if edgeExists.(u,v){
                        if dist[v] > dist[u] + weight(u,v){
                                dist[v] = dist[u] + weight(u,v)

                                parent[v] = u
                        }
                }
        }
}
```