More on Dijkstra's algorithm

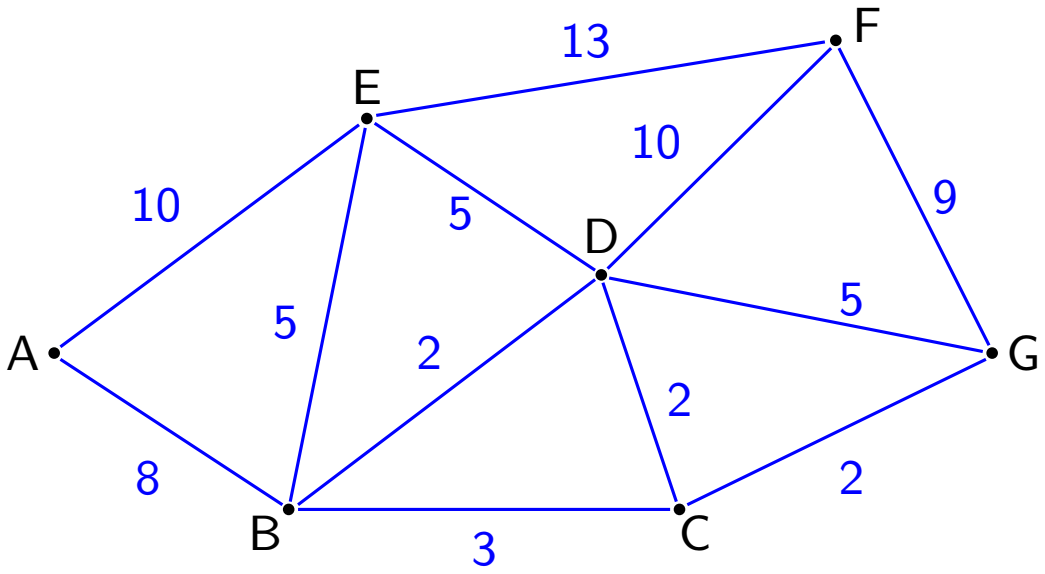
Giorgio Chiovelli and Sebastian Hohmann

20/03/2020

Dijkstra's algorithm

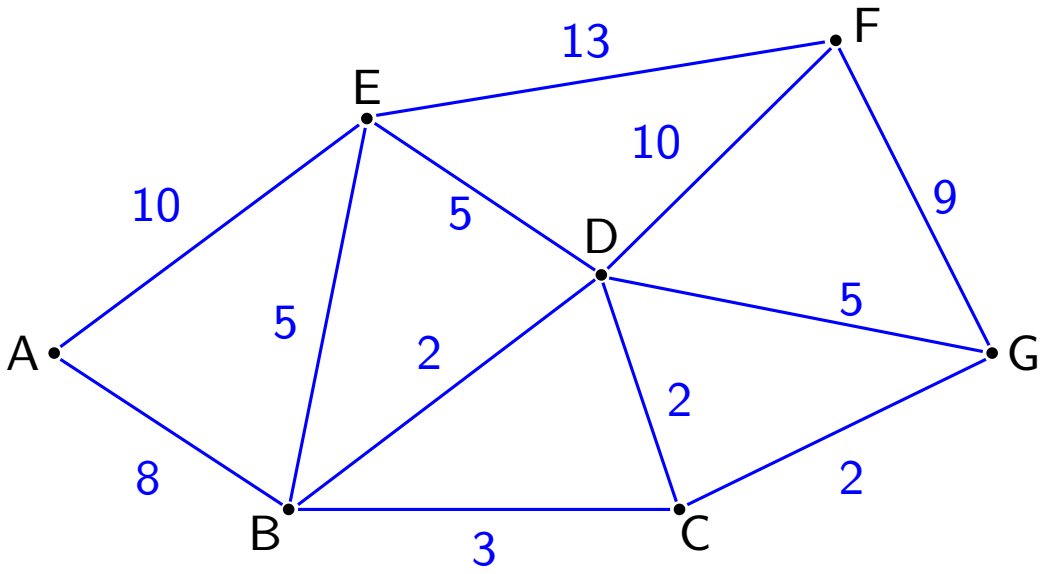
Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.



Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

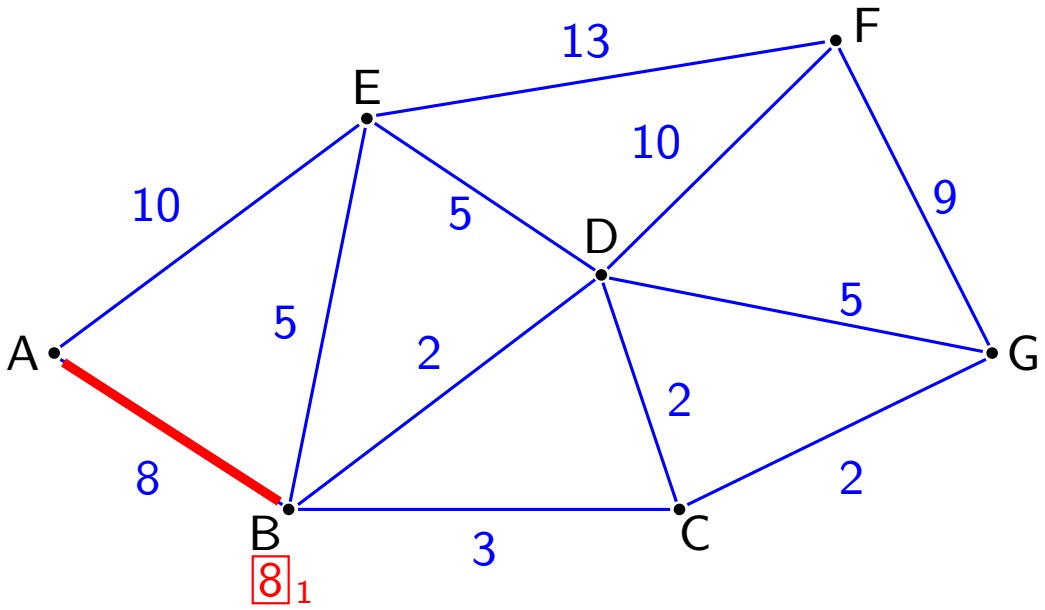


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

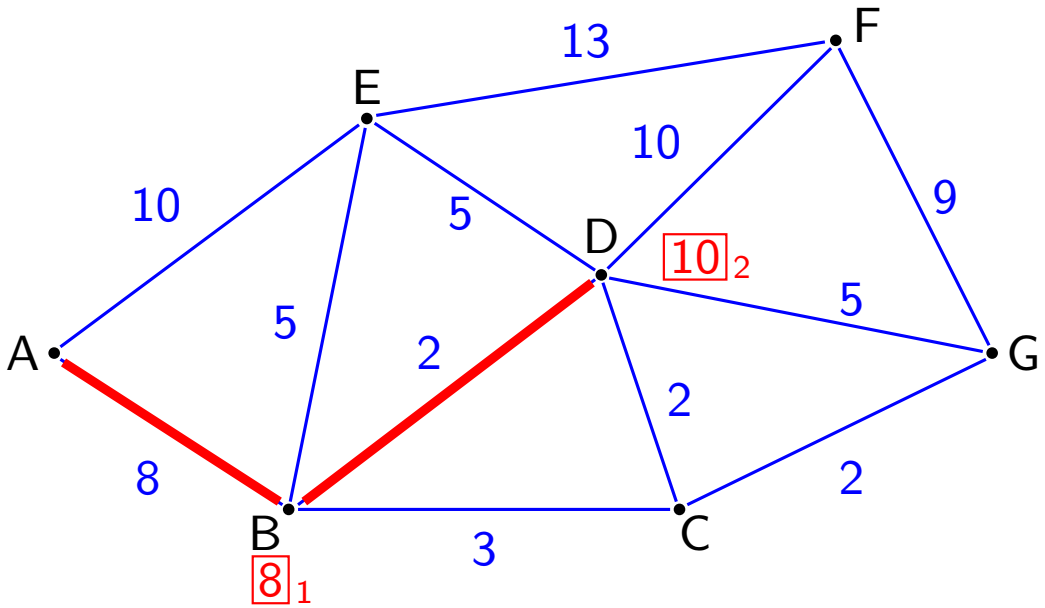


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

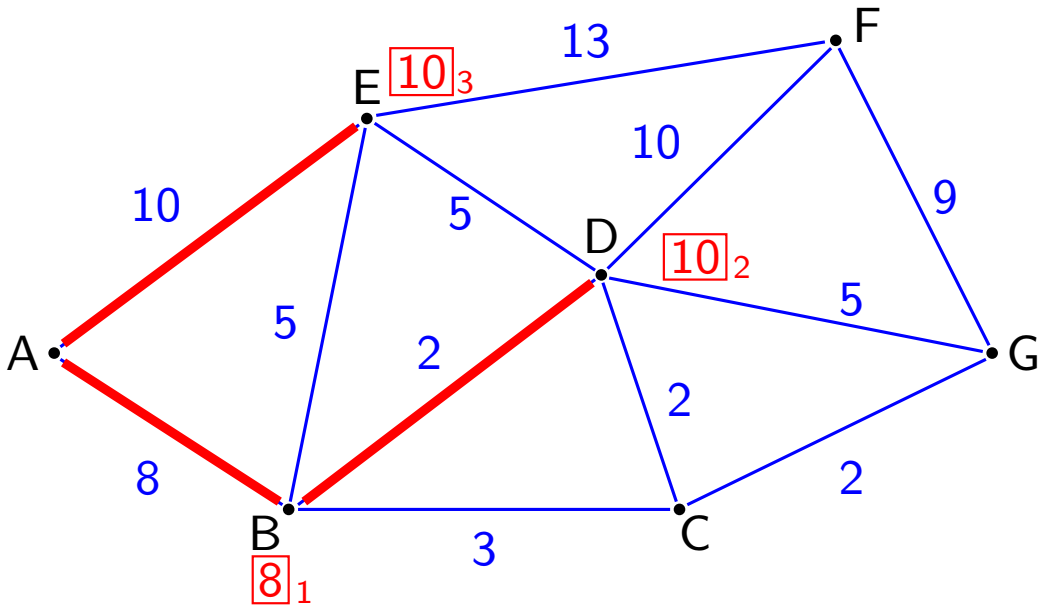


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

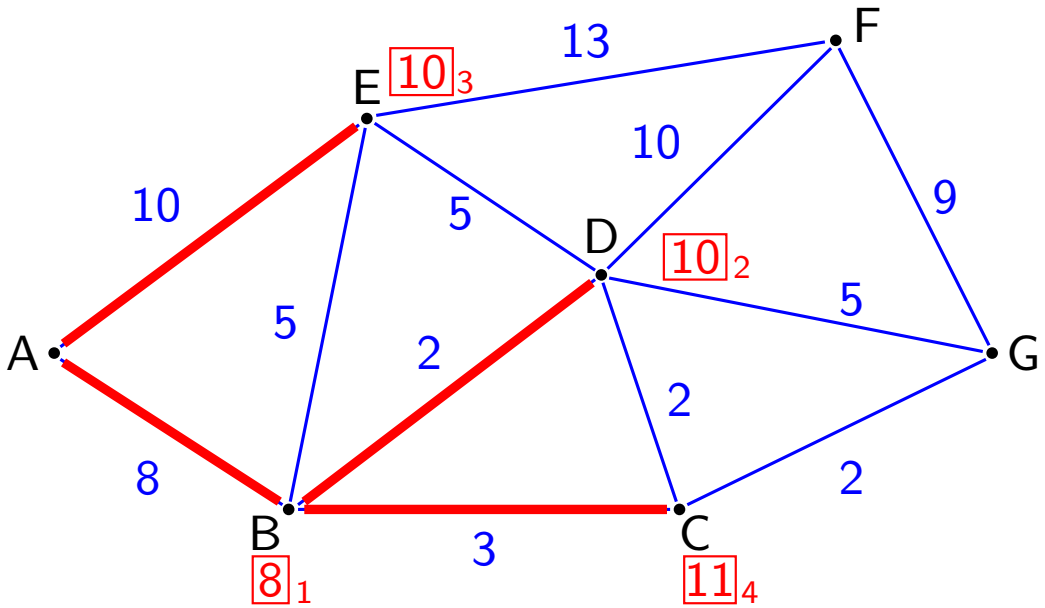


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

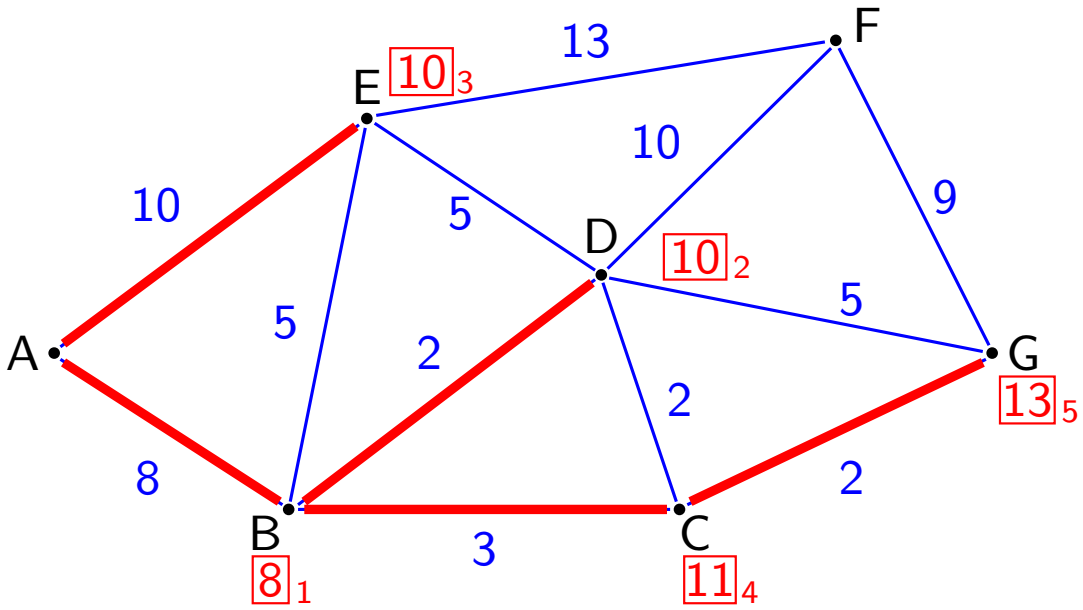


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

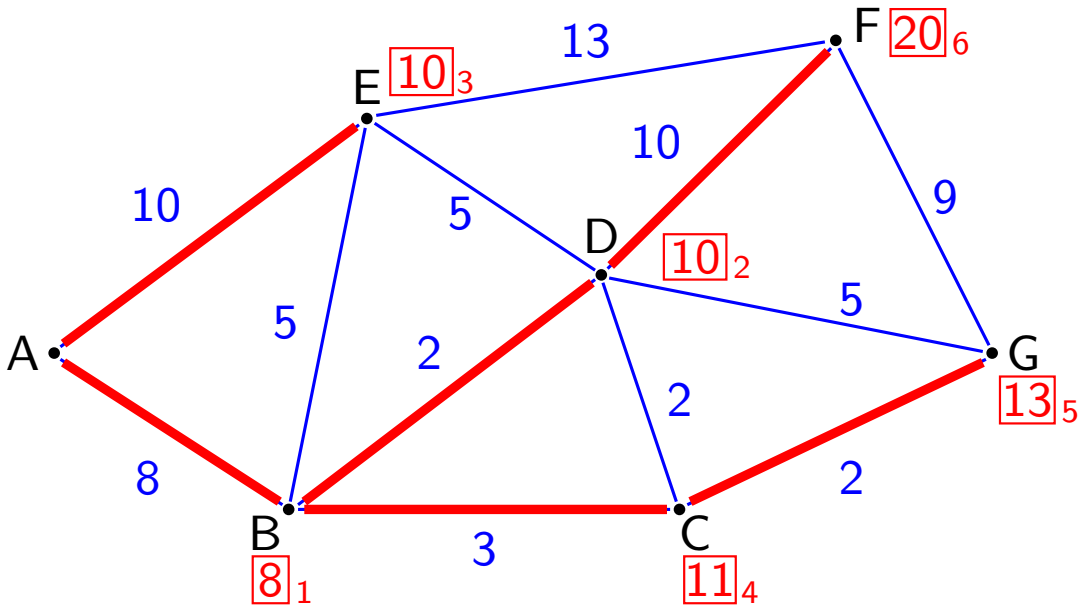


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Shortest path: $ABDF$, **search steps** 6, **distance** 20.



Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

An application

- The basic case (Jackson and Willis, 1995) :

  `https://www.youtube.com/watch?v=6cAbgAaEOVE`

- More containers?

  `http://blancosilva.github.io/post/2016/07/29/decanting.html`