UNIT IV GREEDY TECHNIQUE

Greedy Technique – Minimum Spanning Tree – Prim‘s Algorithm – Kruskal’s Algorithm – Single-source-shortest-paths Problem – Dijkstra’s Algorithm – Huffman Coding – Fractional Knapsack Problem

Shortest paths – Dijkstra's algorithm

***Single Source Shortest Paths Problem*:**

Given a weighted connected (directed) graph G, find shortest paths from source vertex *s* to each of the other vertices

***Dijkstra's algorithm*:**

Similar to Prim's MST algorithm, with a different way of computing numerical labels: Among vertices not already in the tree, it finds vertex *u* with the smallest sum

$$d_v + w(v,u)$$

where

v is a vertex for which shortest path has been already found on preceding iterations (such vertices form a tree rooted at *s*)

d_v is the length of the shortest path from source *s* to *v*

$w(v,u)$ is the length (weight) of edge from *v* to *u*

Notes on Dijkstra’s algorithm

- Correctness can be proven by induction on the number of vertices.

We prove the invariants: (i) when a vertex is added to the tree, its correct distance is calculated and (ii) the distance is at least those of the previously added vertices.

- Doesn’t work for graphs with negative weights (whereas Floyd’s algorithm does, as long as there is no negative cycle)

- Applicable to both undirected and directed graphs

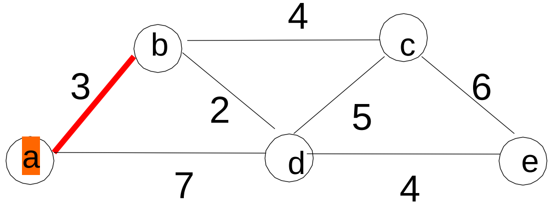
- Efficiency
  - $O(|V|^2)$ for graphs represented by weight matrix and array implementation of priority queue
  - $O(|E|\log|V|)$ for graphs represented by adj. lists and min-heap implementation of priority queue

- Don’t mix up Dijkstra’s algorithm with Prim’s algorithm! More details of the algorithm are in the text and ref books.

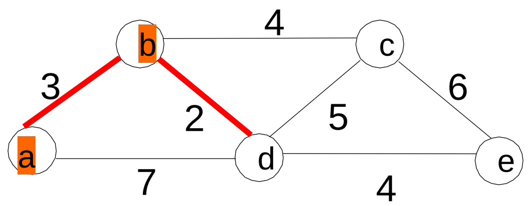
Dijkstra's algorithm Example

Tree vertices | **Remaining vertices**

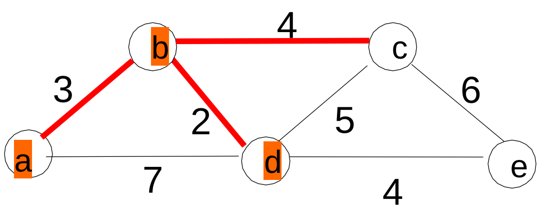
a(-,0) | **b(a,3)** c(-,∞) d(a,7) e(-,∞)



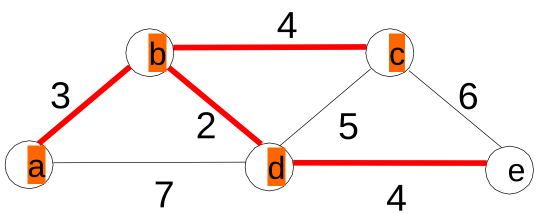
b(a,3) | c(b,3+4) **d(b,3+2)** e(-,∞)



d(b,5) | **c(b,7)** e(d,5+4)



c(b,7) | **e(d,9)**



e(d,9)

Shortest paths – Dijkstra's algorithm

The shortest paths (identified by following nonnumeric labels backward from a destination vertex in the left column to the source) and their lengths (given by numeric labels of the tree vertices) are as follows:

from a to b : a − b of length 3
from a to d : a − b − d of length 5
from a to c : a − b − c of length 7
from a to e : a − b − d − e of length 9

Thank You