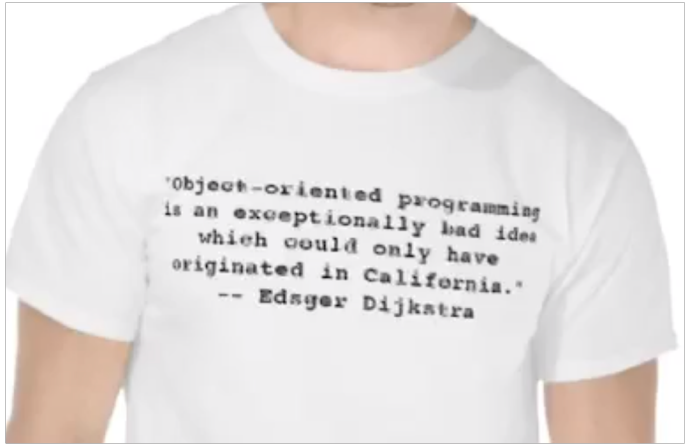
Algorithms and Data Structures

Graphs – Part III – Prim & Dijkstra

Jacob Trier Frederiksen & Anders Kalhauge

Spring 2019

Shortest Paths



1. Path Relaxation
2. Dijkstra's Algorithm – today for simple digraphs only.

The Shortest Paths Problem (single-source)

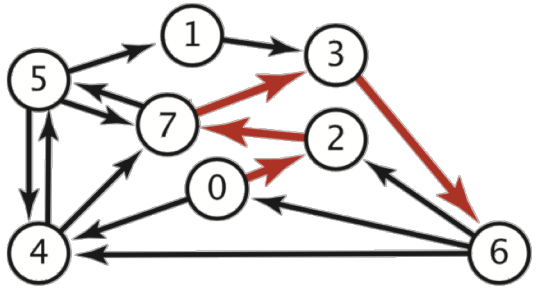
Given a digraph, find the shortest path between a vertex ‘s’ (source) and one other vertex, or several (all) other vertices.

Assume simple digraphs only:

- *Single source*: from one vertex *s* to every other vertex.
- Weights positive
- No negative cycles

edge-weighted digraph

```
4->5  0.35
5->4  0.35
4->7  0.37
5->7  0.28
7->5  0.28
5->1  0.32
0->4  0.38
0->2  0.26
7->3  0.39
1->3  0.29
2->7  0.34
6->2  0.40
3->6  0.52
6->0  0.58
6->4  0.93
```

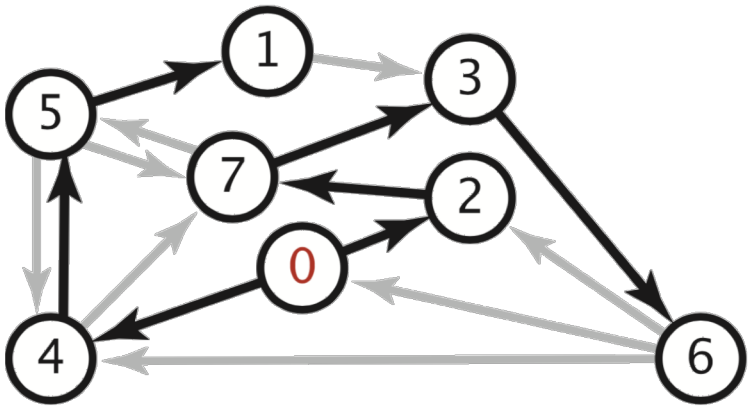


shortest path from 0 to 6

```
0->2  0.26
2->7  0.34
7->3  0.39
3->6  0.52
```

Shortest Path Tree

- Shortest Paths Tree, from computing all single-source paths from ‘s’.
- Represented by two arrays, `edgeTo[]` and `distTo[]`.

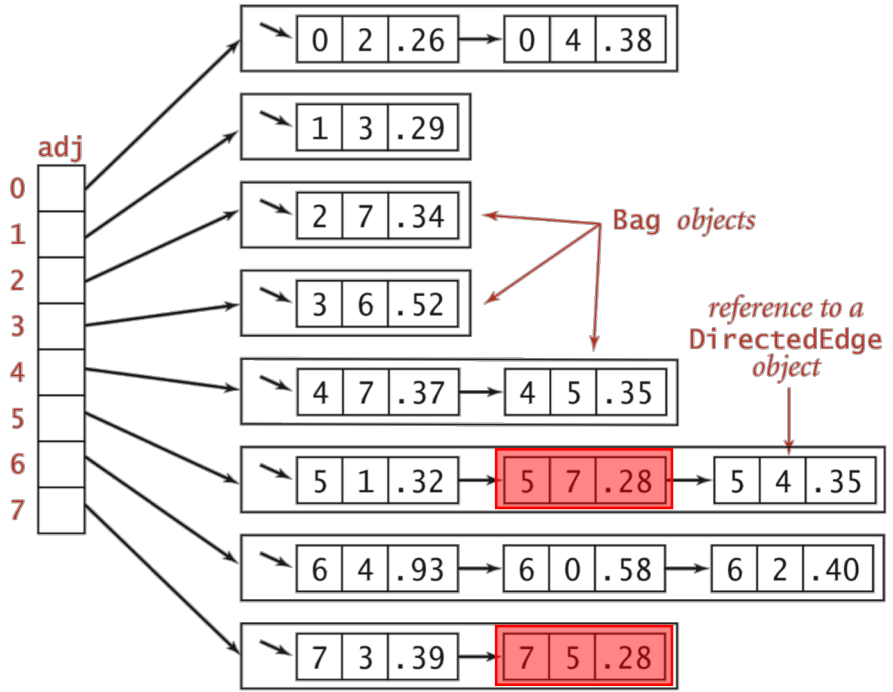


	edgeTo[]	distTo[]
0	null	0
1	5->1 0.32	1.05
2	0->2 0.26	0.26
3	7->3 0.37	0.97
4	0->4 0.38	0.38
5	4->5 0.35	0.73
6	3->6 0.52	1.49
7	2->7 0.34	0.60

What Data Structure for Dijkstra *et al.*?

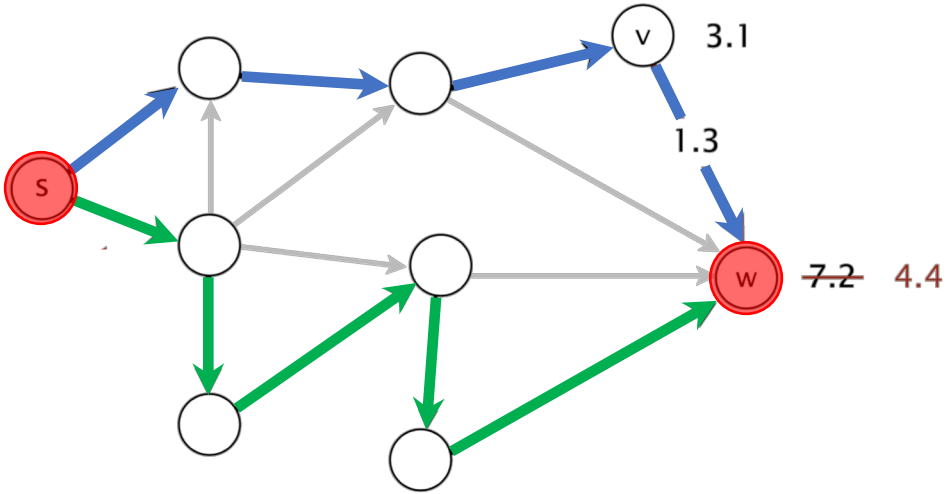
tinyEWD.txt

```
V → 8
    15 ← E
4 5  0.35
5 4  0.35
4 7  0.37
5 7  0.28
7 5  0.28
5 1  0.32
0 4  0.38
0 2  0.26
7 3  0.39
1 3  0.29
2 7  0.34
6 2  0.40
3 6  0.52
6 0  0.58
6 4  0.93
```



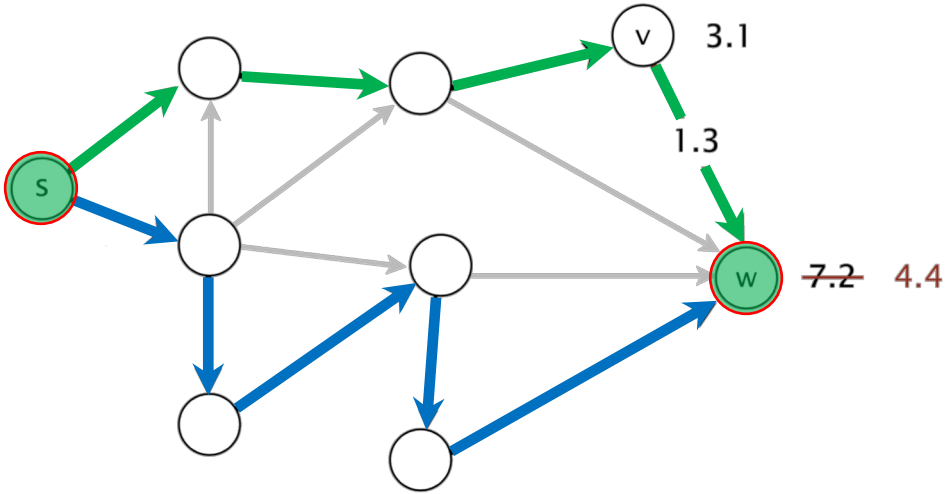
Edge Relaxation

If `e = v → w` gives shorter path to `w` through `v`, update both `distTo[w]` and `edgeTo[w]`.



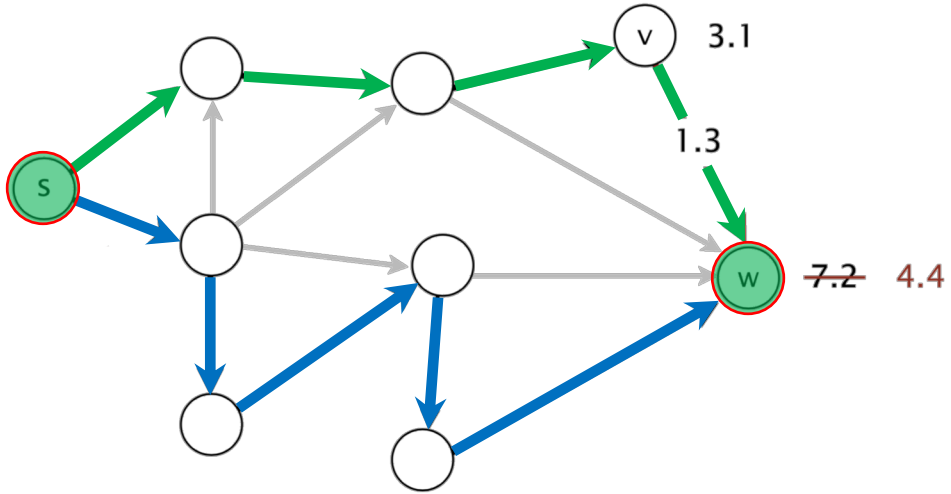
Edge Relaxation

If `e = v → w` gives shorter path to `w` through `v`, update both `distTo[w]` and `edgeTo[w]`.



A Generic SPT Algorithm

1. distTo[s] = 0
2. distTo[v] = ∞
3. While not all optimal:
   Relax edges
4. Done.



Exercise 4

- $e = v \longrightarrow w$ is an edge with weight 17.0.
- At some point, during the generic shortest paths algorithm, `distTo[v]` = ∞ and `distTo[w]` = 15.0. What will `distTo[w]` be after calling `relax(e)`?

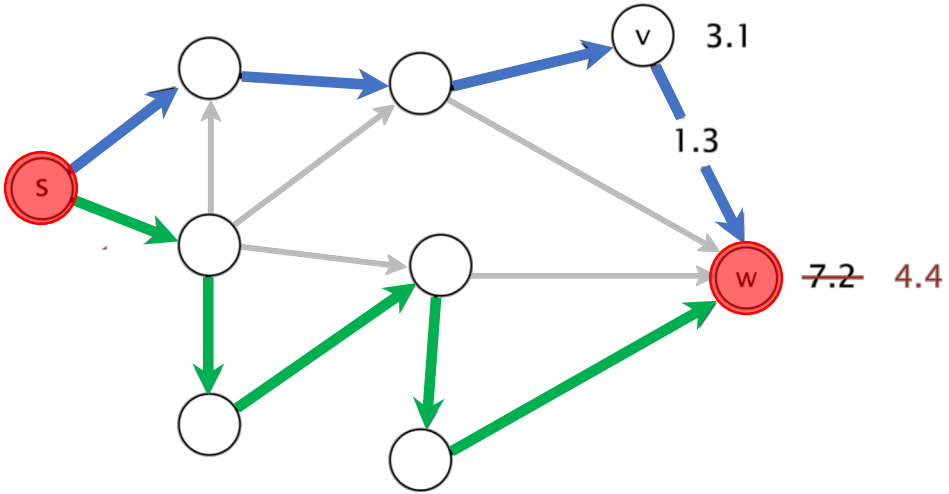
- ❑ The program will encounter a Java runtime exception
- ❑ 15.0
- ❑ 17.0
- ❑ ∞

Edge Relaxation HOWTO?

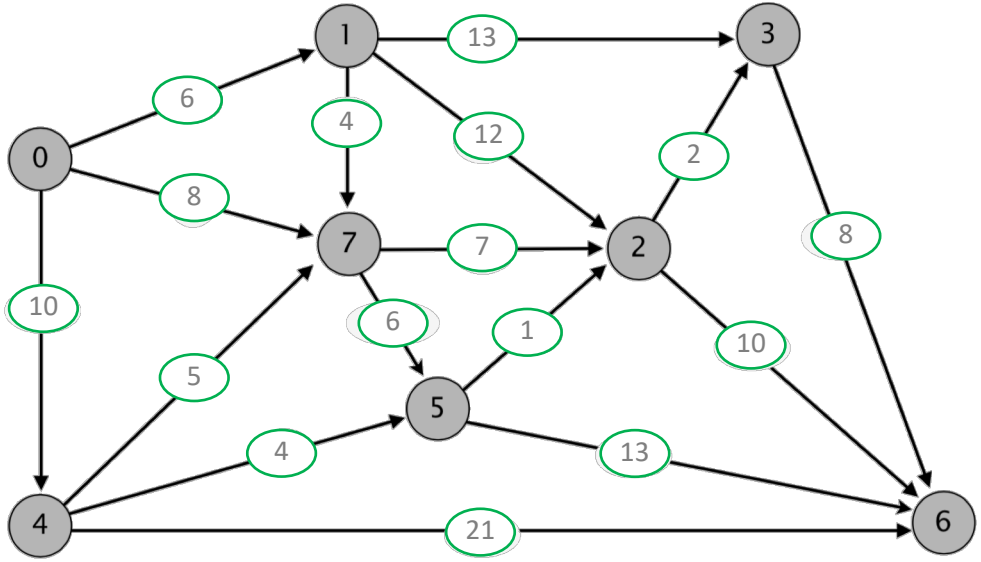
Three central algorithms exist:

1. Dijkstra's Algorithm
2. Topological Sorting
3. Bellman-Ford Algorithm

We might just yet do something about number 2....



Exercise 6: Using Dijkstra's Algorithm



V	distTo[]	edgeTo[]
0	0.0	-
1	6.0	0 → 1
2		
3		
4		
5		
6		
7		

Dijkstra's Algorithm

- Keeping the queued vertices/edges on a minPQ (heap).

PQ implementation	insert	delete-min	decrease-key	total
unordered array	1	V	1	V^2
binary heap	$\log V$	$\log V$	$\log V$	$E \log V$
d-way heap	$\log_d V$	$d \log_d V$	$\log_d V$	$E \log_{E/V} V$
Fibonacci heap	1 †	$\log V$ †	1 †	$E + V \log V$

Exercise 7

For the five example graphs

- `tinyEWG.txt` (V=8, E=16)
- `mediumEWG.txt` (V=250, E=)
- `1000EWG.txt` (V=1000, E=8433)
- `10000EWG.txt` (V=10000, E=61731)
- `largeEWG.txt` (V=1000000, E=7586063)

compare running times, using the `DijkstraSP()` class and the `SP()` class.

Time your computations. You can use fx the Book's `StopWatch()` class for this purpose. Or a system time diff, or you pick...

Exercise 5

Q: What is the BigO running time of Dijkstra's algorithm, using a binary heap?

- V
- E
- V^2
- $E \; Log \; V$

Dijkstra's Algorithm; complexity

- Space: O(*V*)
- Time: O(*E Log V*)

Acknowledgement

- Some material has been taken from the Princeton course ‘Algorithms 4th Edition’, by Sedgewick & Wayne.