

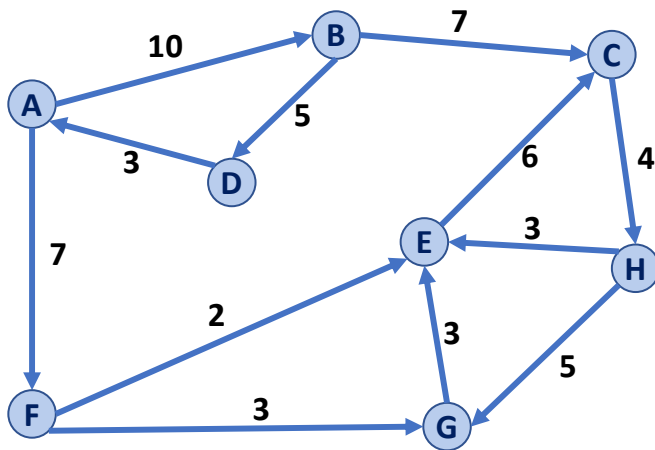
# CS 225

## Data Structures

*December 9 – Delta-stepping (SSSP)*

*G Carl Evans*

# Simplified Delta-stepping Algorithm (SSSP)



```
Delta(G, s, delta):
6  foreach (Vertex v : G):
7  d[v] = +inf
8  p[v] = NULL
9
10 d[s] = 0
11 B[0].push_back(s)
12
13 i = 0
14 while !B.empty()
15     foreach( Vertex v : B[i] )
16         foreach( Edge e : b )
17             relax = cost(u,v) + d[u]
18             if relax < d[v]
19                 d[v] = relax
20                 p[v] = u
21                 B[relax/delta].pushback[v]
22     i = i + 1
```

$\Delta$ -stepping: a parallelizable shortest path algorithm U. Meyer and P. Sanders \*

Type	$p$	k	No. of vertices		
			1M	2M	6M
$\Delta$ -stepping	$1 \cdot 10^{-4}$	60	852	1,770	5,445
Boost Dijkstra	$1 \cdot 10^{-4}$	60	2,423	5,180	16,520
$\Delta$ -stepping	$1 \cdot 10^{-4}$	150	1,402	2,849	10,421
Boost Dijkstra	$1 \cdot 10^{-4}$	150	5,860	11,724	$7 \cdot 10^5$
$\Delta$ -stepping	$1 \cdot 10^{-2}$	60	922	2,026	7,029
Boost Dijkstra	$1 \cdot 10^{-2}$	60	2,984	6,172	19,080
$\Delta$ -stepping	$1 \cdot 10^{-2}$	150	1,887	4,293	16,852
Boost Dijkstra	$1 \cdot 10^{-2}$	150	6,524	13,312	$2 \cdot 10^6$

**Table 2:** Timings in ms on a single core of Xeon E2680v3.  $\Delta$ -stepping was run with  $\Delta = 10$ .

<https://arxiv.org/abs/1604.02113v1>



# Final Exam Review Session

- Implementations
  - Edge List
  - Adjacency Matrix
  - Adjacency List
- Traversals
  - Breadth First
  - Depth First
- Minimum Spanning Tree
  - Kruskal's Algorithm
  - Prim's Algorithm
- Shortest Path
  - Dijkstra's Algorithm
  - Floyd-Warshall's Algorithm

*...and this is just the beginning. The journey continues to CS 374!*