Discussion 4 - Supplemental

CSCI 570

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

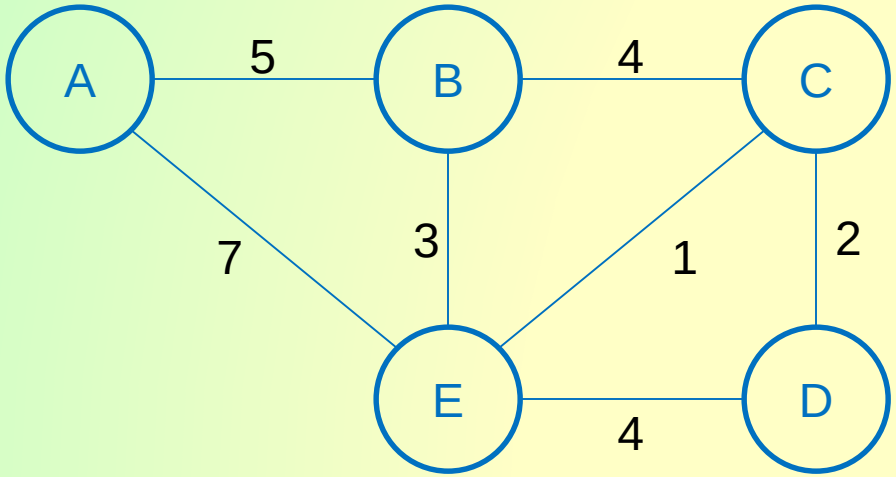
DISCUSSION 4 - SUPPLEMENTAL

Outline

- Problems with Solutions

Problem 1

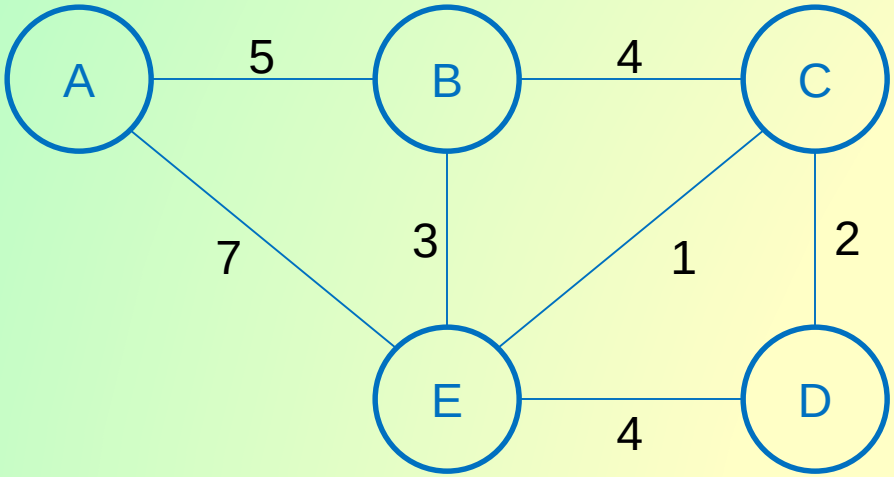
- You are given an undirected graph G with weights $\{w_e\}$ on the edges and the shortest path distances δ(u) from a designated source vertex *s* to every other vertex in G. However, you are not given the actual paths. With this information, give an algorithm to find a shortest path tree from *s* in *O*(E + V) time. Explain why your algorithm correctly computes the shortest paths in the required runtime.
  - The following is just a sample graph to use. You are given the graph with weights and the minimum cost from s=A to all other vertices.



From A	B	C	D	E
Cost	5	8	10	7
Predecessor				

Solution 1

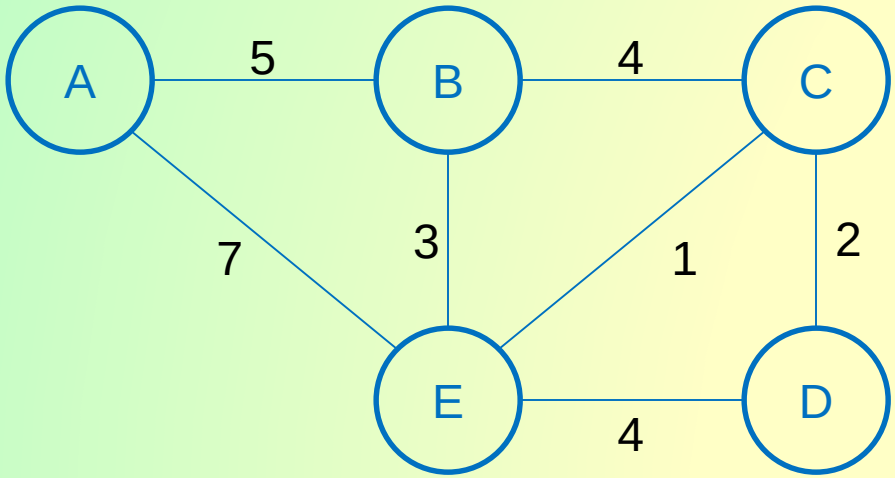
- Dijkstra's Algorithm takes $O(V^2)$ with a min-priority queue and O(E + VlogV) with a Fibonacci heap
  - This isn't quite O(E + V)
- Let's run Dijkstra's Algorithm on the following graph with s=A just to refresh our memory



	B	C	D	E	
Cost	5			7	Expand A
Predecessor	A			A	
	B	**C**	**D**	**E**	
Cost	5	9		7	Expand B
Predecessor	A	B		A	
	B	**C**	**D**	**E**	
Cost	5	8	11	7	Expand E
Predecessor	A	E	E	A	
	B	**C**	**D**	**E**	
Cost	5	8	10	7	Expand C
Predecessor	A	E	C	A	
	B	**C**	**D**	**E**	
Cost	5	8	10	7	Expand D
Predecessor	A	E	C	A	

Solution 1

- Given the graph with the weights and the shortest path cost from s=A to all the other nodes, can we find the predecessors without completely running Dijkstra's Algorithm
- If we start running Dijkstra's Algorithm, we can stop looking at nodes once we know the minimum cost to that node has been found
  - In addition, if we expand a node and find that the cost is greater than the minimum, we don't even need to add it to the table
  - This only requires us to visit each node and each edge exactly one time => O(E + V)



	B	C	D	E	
Cost	5			7	Expand A
Predecessor	A			A	
	B	C	D	E	
Cost	5			7	Expand B
Predecessor	A			A	
	B	C	D	E	
Cost	5	8		7	Expand E
Predecessor	A	E		A	
	B	C	D	E	
Cost	5	8	10	7	Expand C
Predecessor	A	E	C	A	