2019-02-11 Dijkstra's Shortest Path

Monday, February 11, 2019 4:15 PM

- Dijkstra's algorithm allows us to find the shortest path on a weighted graph.
- VERY similar to shortest path on unweighted graph

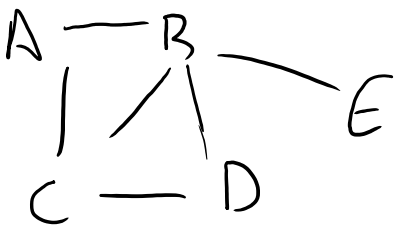
Single Source Shortest Path on Unweighted Graph

(This is a BFS)

1. Enqueue some starting vertex
2. While queue is not empty and destination has not been found
   a. Dequeue from queue, label current
   b. Ask: have we seen current before? If not, enqueue all outgoing vertices

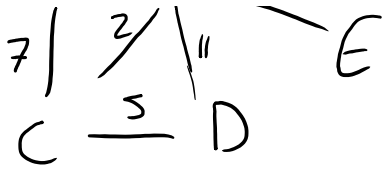
- The only difference between this and Dijkstra's shortest path is that we use a PQ instead of a normal FIFO queue
- When determining distance, we need to track not current distance, but total distance up to that point.

Example on unweighted graph



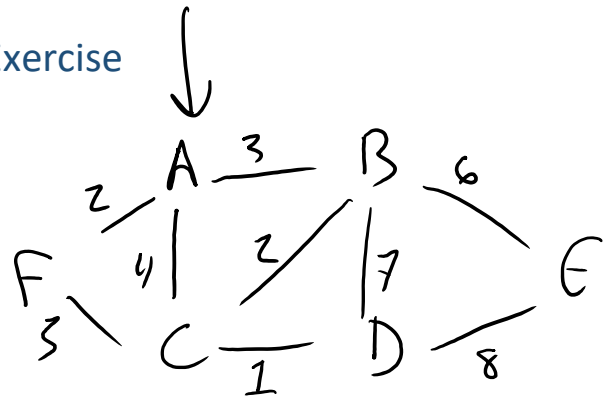
Iteration	Discovered Vertex	State of queue
1		{A:0}
2	A:0	{B:1}, {C:1}
3	B:1	{C:1}, {C:2}, {D:2}, {E:2}
4	C:1	{C:2}, {D:2}, {E:2}, {D:2}
5		{D:2}, {E:2}, {D:2}
6	D:2	{E:2}, {D:2}
7	E:2	{D:2}
8		
9		{Done}

- Same example, this time with weighted graph



Iteration	Discovered Vertex	PQ
1		{A:0}
2	{A:0}	{C:7}, {B:3}
3	{B:3}	{C:7}, {C:5}, {D:14}, {E:4}
4	{E:4}	{C:7}, {C:5}, {D:14}
5	{C:5}	{C:7}, {D:14}, {D:10}
6		{D:10}, {D:14}
7	{D:10}	{D:14}
8		{empty; done}
9		

Class Exercise



Iteration	Vertex Discovered	PQ
1		{A:0}
2	{A:0}	{F:2}, {C:4}, {B:3}
3	{F:2}	{C:4}, {B:3}, {C:5}
4	{B:3}	{C:4}, {C:5}, {C:5}, {E:9}, {D:10}
5	{C:4}	{C:5}, {C:5}, {E:9}, {D:10}, {D:5}
6		{C:5}, {E:9}, {D:10}, {D:5}
7		{E:9}, {D:10}, {D:5}
8	{D:5}	{E:9}, {D:10}, {E:13}
9	{E:9}	{D:10}, {E:13}
10		{E:13}
11		{empty}