

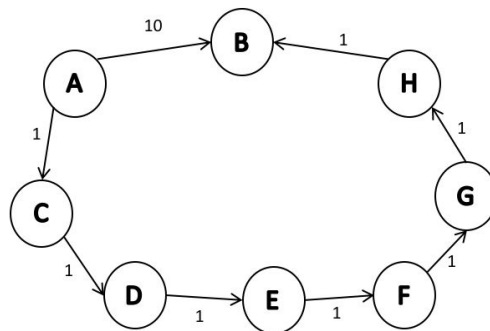
CS 225 Spring 2019 :: TA Lecture Notes

4/26 Graph problems

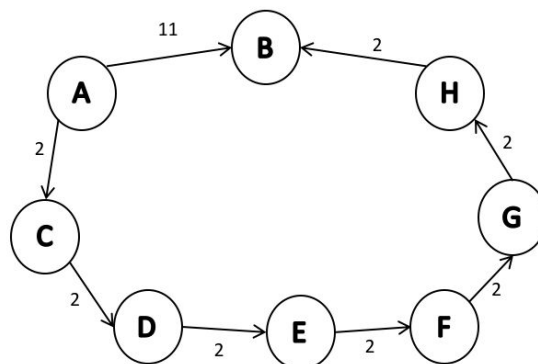
By Wenjie

- **Dijkstra's algorithm performance and edge cases / SSSP**

- Dijkstra gives us the shortest path from our path (single source) to every connected vertex!
- Q: How does Dijkstra handle a single heavy-weight path vs. many light-weight paths?



- It will pick the path A-C-D-E-F-G-H-B instead of A-B because the first path has length 7 and the second path has length 10.
- If we want to get the most direct path instead of the shortest path, we can adjust edge weights.
 - For example, we can add 1 to all edges. In that case, path A-C-D-E-F-G-H-B will be of length 14, while path A-B will be 11 and Dijkstra would pick A-B.

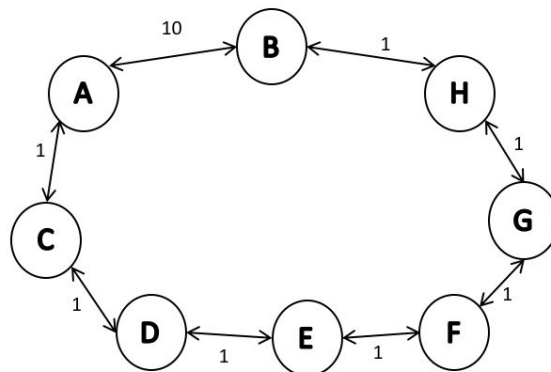


CS 225 Spring 2019 :: TA Lecture Notes

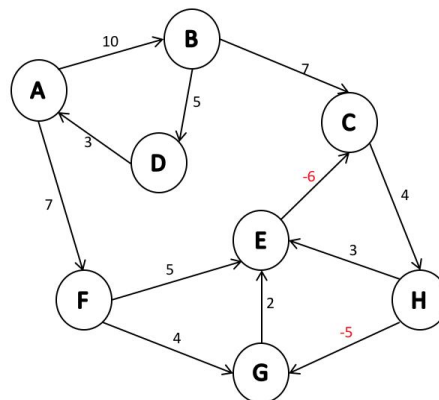
4/26 Graph problems

By Wenjie

- When there is a tie in path lengths, it is up to us to decide how we want to handle that.
- Can Dijkstra's algorithm handle undirected graphs?



- Yes, it can. It will not go back or in loop because that will increase the path length.
- Can Dijkstra's algorithm handle graph with negative cycles?



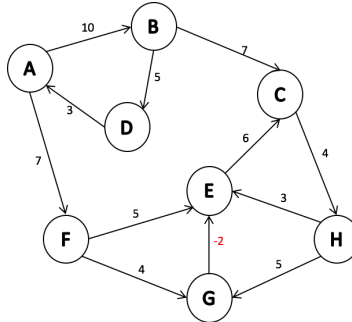
- No, because negative weight cycle doesn't have defined shortest path. We can always find a shorter path which leads to negative infinity.
- Dijkstra's algorithm can handle graphs with negative edges, but no negative cycles - it will finish, there will be no infinite loop. However, it will not produce the shortest path.

If we run Dijkstra on the following graph:

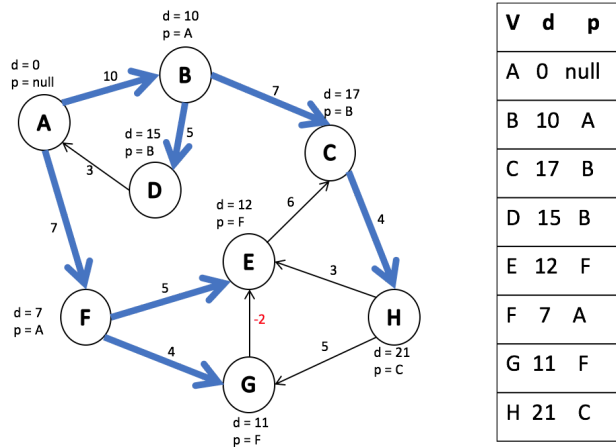
CS 225 Spring 2019 :: TA Lecture Notes

4/26 Graph problems

By Wenjie



We will obtain the following result if we starts at A:



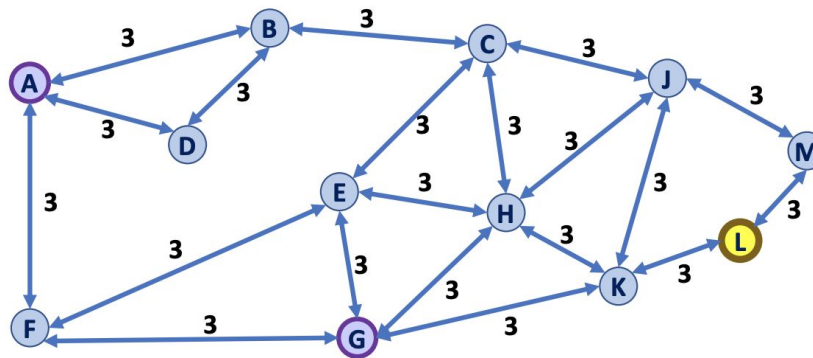
However, we can see that path A-F-E is longer than path A-F-G-E.

- Negative cycles are limitation in math, while inability to find shortest path when there is a negative edge (and no negative cycle) in the graph is a limitation of Dijkstra's algorithm.
- What if we have a 0 edge?
 - Dijkstra works correctly with 0 edges.
- Running time of Dijkstra's algorithm
 - Remember, we built Dijkstra's algorithm on top of Prim's algorithm.
 - We only added two lines of code which take $O(1)$.
 - Therefore, Dijkstra's running time is the same as Prim's.
- Landmark path problem

CS 225 Spring 2019 :: TA Lecture Notes

4/26 Graph problems

By Wenjie



Suppose you want to travel from A to G.

Q1: What is the shortest path from A to G?

Q2: What is the fastest algorithm to use to find the shortest path?

In your journey between A and G, you also want to visit the landmark L.

Q3: What is the shortest path from A to G that visits L?