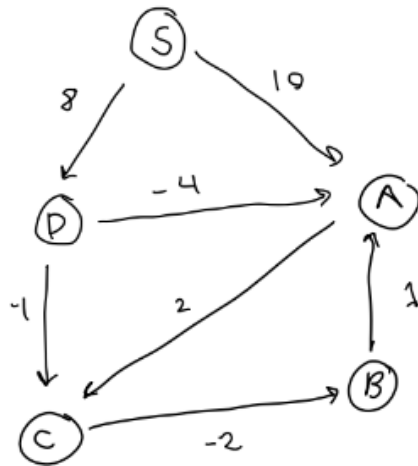

1: Basics

Fill in the definition of each of the following.

1. Bellman-Ford:

Find the shortest path for the following graph using:

1. Bellman-Ford



2: Practice Problems

1. The Bellman-Ford algorithm can take $O(|V||E|)$ time because each edge can be relaxed up to $|V| - 1$ times without finding a negative cycle. Construct a family of directed, weighted graphs, where edge weights can be positive, negative (or even 0) where some specific edge e is relaxed $\Omega(|V|)$ times in a valid run of Bellman-Ford.

Hint: the solutions will give an example of a construction where a specific edge is relaxed exactly $|V| - 2$ times, but you just need a family of graphs such that some edge is relaxed a number of times proportional to the number of nodes.

Helpful Videos:

- Bellman-Ford in 5 minutes — Step by step example, Micheal Sambol, https://youtu.be/obWXjtg0L64?si=_mHlxtr3yMLP1FLG

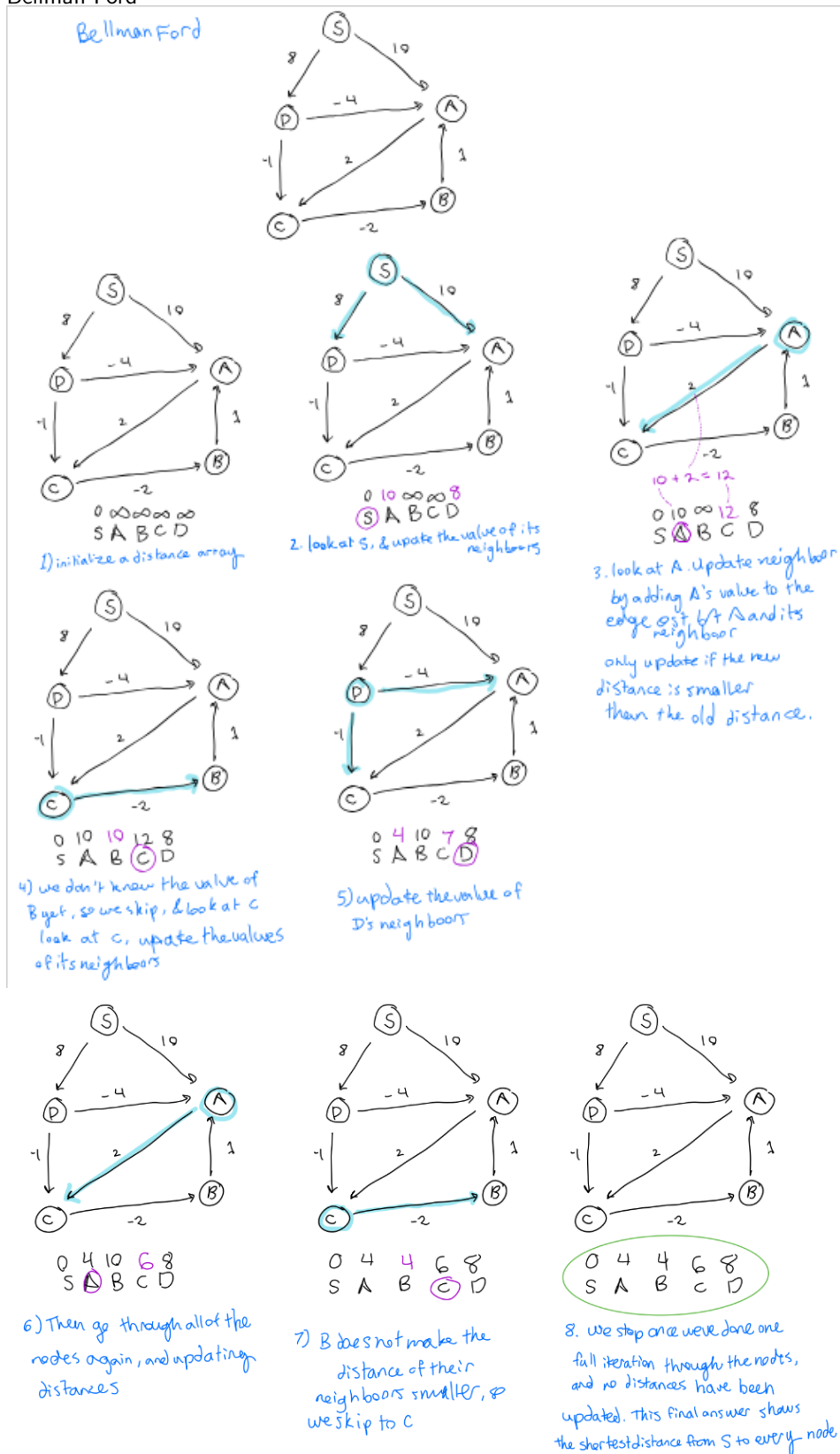
3: Solutions

Fill in the definition of each path finding algorithm.

1. Bellman-Ford: The goal is to find the shortest distance from our starting node s to every other node, in a graph that contains negative edge weights. It uses the same relaxation rule as Dijkstra, except will iterate over the nodes multiple times until the distances no longer update.

Find the shortest path for the corresponding graph :

1. Bellman-Ford



4: Practice Problems

1. We will construct a fan graph as follows: first stack $n - 2$ nodes in a directed line graph, starting with $s = v_{n-2}$, which points with an edge of weight 0 to v_{n-3} , pointing at v_{n-4} , all the way down to v_1 . Then point an edge from each v_i to a new node u with weight i . Point an edge from u to the target node t with weight 0. Then in Bellman-Ford, the edge (u, t) will be relaxed $n - 2$ times, since $d[u]$ keeps going down from $n - 2$ in decrements of 1, and each time, you can then relax the (u, t) edge again.

