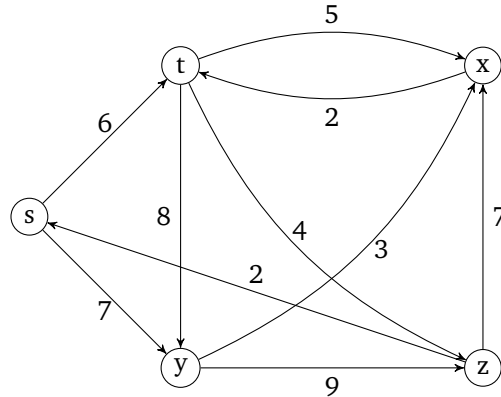


Section 08: Graph traversal, shortest path algorithms

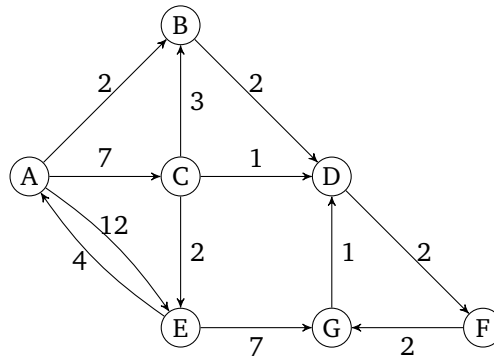
1. Simulating Dijkstra's

(a) Consider the following graph:



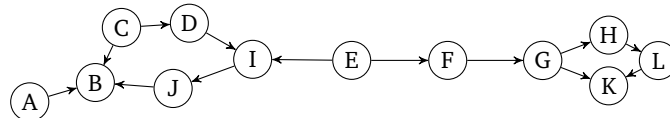
Suppose we run Dijkstra's algorithm on this graph starting with vertex s . What are the final costs of each vertex and the shortest paths from s to each vertex?

(b) Here is another graph. What are the final costs and shortest paths if we run Dijkstra's starting on node A ?



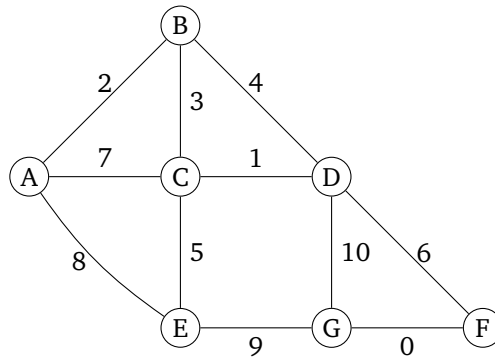
2. Practicing topological sort

Find a topological sort of the following graph:

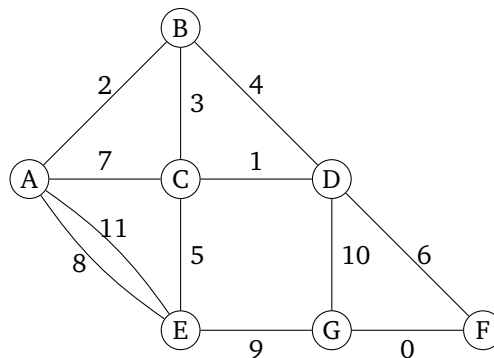


3. Minimum spanning trees

Consider the following graph:

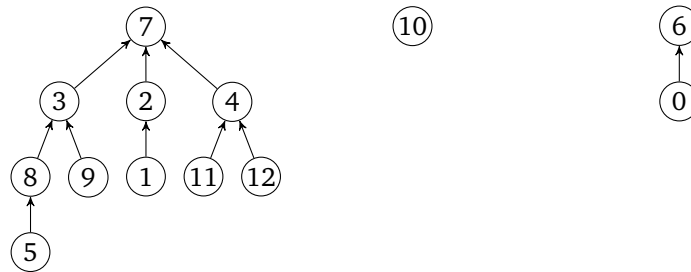


- What happens if we run Prim's algorithm starting on node A ? What are the final costs and edges selected? Give the set of edges in the resulting MST.
- What happens if we run Prim's algorithm starting on node E ? What are the final cost and edges selected? Give the set of edges in the resulting MST.
- What happens if we run Prim's algorithm starting on *any* node? What are the final costs and edges selected? Give the set of edges in the resulting MST.
- What happens if we run Kruskal's algorithm? Give the set of edges in the resulting MST.
- Suppose we modify the graph above and add a heavier parallel edge between A and E , which would result in the graph shown below. Would your answers for above subparts (a, b, c, and d) be the same for this following graph as well?



4. Disjoint sets

Consider the following trees, which are a part of a disjoint set data-structure:



For these problems, use both the **union-by-rank** and **path compression** optimizations. Assume that the first tree has rank 3, the second has rank 0 and the last has rank 1.

- Draw the resulting tree(s) after calling `findSet(5)` on the above. What value does the method return?
- Draw the final result of calling `union(2,6)` on the result of part a.

5. Design Problem: Pathfinding in mazes

Suppose we are trying to design a maze within a 2d top-down video-game. The world is represented as a grid, where each tile is either an impassable wall, an open space a player can pass through, or a *wormhole*. On each turn, the player may move one space on the grid to any adjacent open tile. If the player is standing on a wormhole, they can instead use their turn to teleport themselves to the other end of the wormhole, which is located somewhere else on the map.

Now, suppose there are several coins scattered throughout the map. Your goal is to design an algorithm that finds a path between the player and some coin in the fewest number of turns possible.

Describe how you would represent this scenario as a graph (what are the vertices and edges? Is this a weighted or unweighted graph? Directed or undirected?). Then, describe how you would implement an algorithm to complete this task.

6. Design Problem: Negative Edge Weights

If you enjoy reading Pokémon jokes, you can read the flavor text to understand where the graph problem comes from. Otherwise, you can skip those parts and just read the formal statements.

Flavor Text: You and your trusty Pikachu have made it halfway through Viridian Forest, but things have taken a turn for the worst. That last Weedle poisoned your Pikachu, and you're all out of antidotes.

In the Pokémon world, the poison doesn't do any damage as long as you stay *perfectly still*. But every time you take a step, the poison does a little bit of damage to your poor friend Pikachu.

Thanks to Bulbapedia¹, you know the exact map of Viridian Forest. Knowing that each step will cost your Pikachu exactly one of its precious hit points, you will need to find an efficient path through the forest.²

Formal Statement: In a video game you are playing, each step you take costs a character (Pikachu) one unit of health. You have a map of the level (Viridian Forest) – your goal is to reach the end of the level (marked on your map) while losing as little health as possible.

¹Like Wikipedia, but for Pokémon!

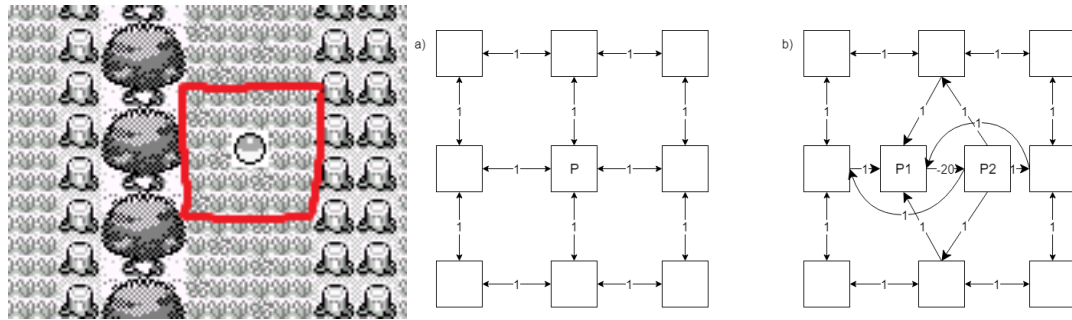
²Don't worry about running into wild Pokémon. For some reason you have a huge number of repels. Next time, maybe invest in full heals or potions instead.

- (a) Describe a graph and an algorithm run on that graph to find the path through the forest to save as many of Pikachu's hit points as possible (i.e. the path with the fewest number of steps).
- (b) **Flavor Text:** You run your algorithm and come to a devastating realization – the edge of Viridian Forest is at least 25 steps away, and Pikachu has only 20 hit points left. If you just walk to the end of the forest, Pikachu will faint before reaching the next Pokémon Center. So you come up with a backup plan. Returning to Bulbapedia, you see there is a potion just a little bit out of the way of the fastest path.

Brock tells you he knows how to update your graph to find the best path now. He says he'll add a dummy vertex to the graph where the potion is and connect up the new vertex with a (directed) edge of length -20 , to represent undoing the loss of 20 hit points.

Formal Statement: You realize your character doesn't have enough health to make it to the edge of the forest. But you know there is a healing item (a "potion") somewhere in the forest, that will give you back 20 units of health.

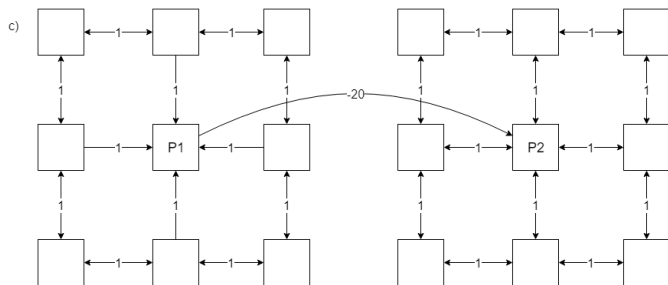
A friend (Brock) suggests the following update: add a dummy vertex to the graph where the healing item is and connect up the new vertex with a (directed) edge of length -20 , to represent undoing the loss of 20 hit points.



9 spots in Viridian Forest, the corresponding vertices before Brock's transformation and the same vertices after the transformation.

Tell Brock why his representation isn't quite going to work (hint: you can only use the potion once. What happens if the potion edge is part of a cycle?).

- (c) You convince Brock to change the graph representation. You'll now have two copies of the original Viridian Forest graph, in copy 1 the potion is still unused. In copy 2, the potion is no longer there. You add an edge of weight -20 from copy 1 to copy 2 at the location of the potion (crossing that edge represents using that potion). His new graph looks something like this.



Brock says he'll start running Dijkstra's. Should you trust the output?

- (d) **Challenge Problem:** Misty says she knows about another potion over there somewhere. Describe how to modify the graph to handle both of the potions.