1. As Bellman-Ford runs, it repeatedly loops over the edges in a graph. In class we saw that it may need to loop over the list of edges $n - 1$ times. In this problem we will see a way to reduce the worst case by a constant factor. Throughout this problem we will assume the Bellman-Ford algorithm starts at vertex $a$. Consider the following graph:

(a) What ordering of the edges will maximize the number of loops through the list of edges that Bellman-Ford performs? What ordering minimizes the number of loops? How many loops are performed in each case?

(b) Suppose we assign numbers from 1 to $n$ to the vertices in the graph. Then if an edge goes from a lower number to a higher number, color it red. If an edge goes from a higher number to a lower number, color it blue. For this new version of Bellman-Ford, we will order the edges as follows: the red edges out of vertex 1, the red edges out of vertex 2, ..., the red edges out of vertex $n$, the blue edges out of vertex $n$, the blue edges out of vertex $n - 1$, ..., and finally the blue edges out of vertex 1. Labeling the vertices with their numerical equivalent, color the edges in the following graph and run Bellman-Ford using this edge orderings.
(c) Argue that if a shortest path is made up of only red edges, Bellman-Ford will compute that shortest path’s distance in one round. Similarly argue that if a shortest path is made up of only blue edges, Bellman-Ford will compute that shortest path’s distance in one round. Hint: it suffices to show the edges of a shortest path are relaxed in order.

(d) From part (c), argue that if a shortest path is multi-colored and starts with a red edge, then the number of rounds required is one plus the number of times it switches from blue to red. For example if the edges on a five edge shortest path are red, red, blue, blue, red, then Bellman-Ford with the special edge ordering takes two rounds to compute that path’s length (plus a final loop to verify that nothing changes).

(e) How many times can a shortest path of \( k \) edges that starts with a red edge switch from blue to red?

(f) Knowing that the longest possible shortest path contains \( n-1 \) edges, how many rounds could Bellman-Ford take to compute its distance? Conclude that this alternate version of Bellman-Ford requires at most \( n/2 \) loops through the list of edges.

This improvement is called Yen’s improvement. Professor Eppstein (a UCI Computer Science professor) and Professor Bannister (a former UCI graduate student and current Santa Clara University professor) found a way to further speed up Bellman-Ford to roughly \( n/3 \) loops through the list of edges.

2. Run the following code:

```python
make_set(1);
make_set(2);
make_set(3);
make_set(4);
make_set(5);
union(1,2);
union(1,3);
union(4,5);
union(1,4);
find(5);
```

What are the parents and ranks of each element after each union and find call?
3. In a union find data structure, call `make_set` 15 times and perform some union operations. How long of a chain in the parent map can you form when the union find data structure does not use union-by-rank or path-compression? How long of a chain can you form when it does?

4. Sometimes people mistakenly believe the edges of a minimum spanning tree are also the edges of a shortest path originating from some vertex. Show that this is sometimes the case by drawing a graph and indicating a vertex such that the shortest paths out of that vertex form the minimum spanning tree of the graph. Next show that it does not always hold by drawing a graph such that the minimum spanning tree is not the tree of shortest paths out of any vertex.