CSC 226 - Algorithms and Data Structures II

Assignment 3

Ben Wolfe — V00205547

Problem 1

A modified version of the Bellman-Ford algorithm provides an efficient way for finding negative cycles in a graph. The longest possible path, **without a cycle**, that can exist in a graph is $|V| - 1$ edges. For this reason, original Bellman-Ford algorithm scans the edges $|V|-1$ times to ensure the shortest path has been found. If an **additional scan** is completed and the values change, then it means a path of length $|V|$ edges has been found which can **only** occur if a negative cycle exists in the graph.

The algorithm is broken up into three steps:

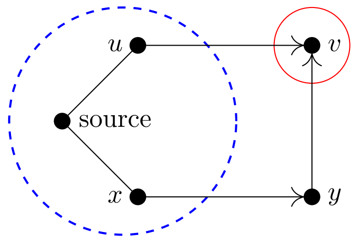
- First, use the Bellman-Ford algorithm for $|V| - 1$ iterations
- Second, apply one more iteration of the algorithm to identify cycles by tracking which values change. The vertices that change are vertices in a negative cycle.
- Trace back through the predecssor tree to find the other verticies in the cycle.

The pseudocode for this algorithm is shown below:

```
procedure FindNegativeCycles(vertices, edges, source)

  distance[] ← Empty array
  predecessor[] ← Empty array

  for each v in vertcies do
    distance[v] = ∞
    predecessor[v] = null
  end for

  distance[source] = 0
  for i = 0, 1, ..., |V| − 1 do
    for u = 0, 1, ..., |V| do
      for Edge (u,v) with weight w in v.adjacentEdges do
        if distance[v] > distance[u] + w then
          distance[v] = distance[u] + w
          predecessor[v] = u
      end for
    end for
  end for

  visitedCycleVertices ← Empty array of size |V|
  for u = 0, 1, ..., |V| do
    for Edge(u,v) with weight w do
      if distance[v] > distance[u] + w then
        *Being in this loop means that a negative cycle exists*
        while predeccessor[v] ≠ starting vertex and vistitedCycleVertices ≠ visited do
          Print v
          visitedCycleVertices[v] ← Mark as visited
        end while
      end if
    end for
  end for
end procedure
```

Problem 2

The correctness of Dijkstra's algorithm is verified by proof of contradiction. The goal is to show that a path through y to get to v leads to a contradiction indicating that the shortest path to v **must** be directly from the established set of points (shown in the dotted blue circle).



While making this proof, we make one **key** assumption:

- Everything in the set is already the closest distance to the source.

In our proof, we state that the best possible distance of x (BPD[x]) is equal to d[x]. In other words, the points in the blue cloud are *already* considered optimum. This assumption is void when considering negative edges since a **new** *indirect* path to x may exist.

Problem 3

Claim: If G is even, then G has a cycle decomposition.

Proof (by strong induction):
Basis
$|E(G)| = 0$ then $\mathcal{S} = \{\}$ - This corresponds to a trivial decomposition (empty set) so the claim holds.

Inductive Hypothesis
Suppose that for all even graphs on $< m$ edges, there exists a cycle decomposition.

Inductive Step

- Take G to be even and to have m edges ($|E(G)| = m$)
- Since the graph is even, all vertices will either have a multiple of two edges coming out of them, or no edges.
- Let X be the set of vertices with degree > 0.
- Next, $F = G[X]$ where F is the graph induced on the vertices with degree > 0.
- F is an even graph and has vertices with degree ≥ 2, therefore F must contain a cycle.

Theorm: Let G be a graph in which all vertices have degree ≥ 2, then G contains a cycle.

- Let P be the longest path in G $(v_0, v_1, ..., v_{k-1}, v_k)$
- Since $\deg(v_k) \geq 2$, there is a vertex $v \in V(G)$ that is not v_{k-1} that is adjacent to V_k
- If v does not belong to the path P, then we have a longer path in G call it P' $(v_0, v_1, ..., v_{k-1}, v_k, v)$
- This is impossible since we said the P is already a longest path
- Then, v must be one of the vertices on path P; $v = v_i$ for some i where $0 \leq i \leq k-2$
- Then $v_i, v_{i+1}, ..., v_{k-1}, v_k$ is a cycle in G

Now, back to the proof of cycle decomposition:

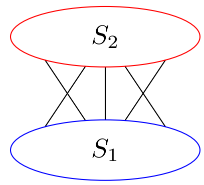
- We've established that there is a cycle in F, call it C
- Take $G' = G \setminus E(C)$
- G' is an even graph with $< m$ edges. Therefore, by the induction hypothesis, G' has a cyle decomposition C'.
- Therefore, $\mathcal{C} = C' \cup C$ is a cycle decomposition of the original graph G. $\square$

Problem 4

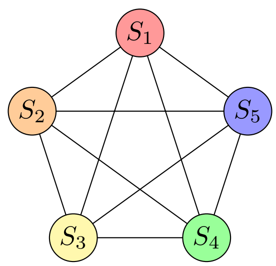
Part (a)

The graph consists of independent sets, or color classes, $V_1, V_2, ..., V_5$. Therefore, the chromatic number of G will be **5** which can also be verified in the following way:

We know that any two pairs of vertices form a complete bipartite graph.

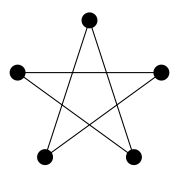


Since this holds for every pair of vertices, a simplified graph can be drawn showing the relationship between all verticies. A single edge here denotes the multiple edges shown above.



Part (b)

The graph is not 2-colorable due to the fact it contains an odd cycle. From *Konig, 1936:* A graph is bipartitate, i.e. 2 colorable, **if and only if** it does not contain an odd cycle. Both the inner star and outer pentagon violate this theorem.



A valid 3-coloring is shown below: