Assignment 5 Problem 1

Rather than solving the problem in graph G, we need to work on an extended graph G’ of G.

To construct G’, we do the following thing:

* For each vertex v in G, there are k corresponding vertices in G’ namely
* For each edge uv in G, there are k corresponding edges in G’ namely

( i.e. )

By doing this, we have another graph G’ with kn vertices and km edges.

Note that constructing G’ costs O(km+kn) time.

Then just doing an BFS on G’ with source vertex , then the level number of in the BFS tree is the shortest length of walk from s to t whose length is divisible by k in G.

To prove the correctness, here is the reason why constructing G’ by the way that stated previously. For each vertex , this means that the walk reaches with walk length . And each edge (for ) means that at the end of a walk length , we walk 1 length further, then the new walk is of length . And for edge , this means that at the end of a walk length , where , we walk 1 length further, then the new walk is of length . Then by constructing the G’, we have initially add some modulo information from G into G’.

The reason to let as the source vertex in BFS is because we want to use as a source vertex in G, the length of the shortest walk from G to G is obviously 0. Because we want to find the shortest walk from s to t whose length is divisible by k, so as long as we reach in G’, we have a walk that is divisible by k by the reason stated previously (the way we construct edges force the correctness). And since the level number of BFS tree for each vertex is minimized. Therefore, the level number of in the BFS tree is the shortest walk which is divisible by k.

The running time for BFS on a graph with kn vertices and km edges is O(kn+km).

So, the total running time for this algorithm is .

Since we assume that , the runtime for the algorithm is