CSDS 455: Homework 2

Shaochen (Henry) ZHONG, `sxz517`

Due and submitted on 08/31/2020
CSDS 455, Dr. Connamacher

1 Problem 1

Arbitrarily label the given graph G as following. For the ease of description, let the LHS of the graph to be set U, and the RHS to be set V. We will also have an empty list M to store matching.

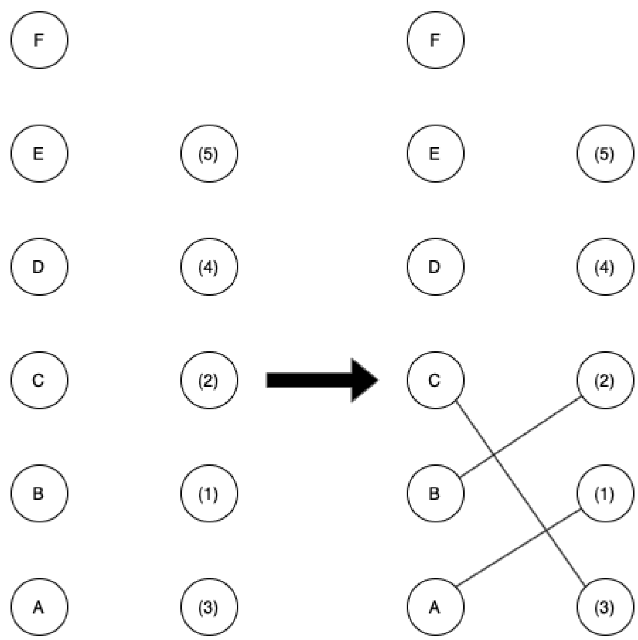

Figure 1: Labelled G and its trace after first 3 iterations

As a rule of this `BFS`, the algorithm will try to match the node with lowest alphabetical value in U to the node with lowest numerical value in V. Thus, the algorithm will check on edges $< A, (1) >$, $< B, (2) >$, and $C, (3)$ respectively. Since all of them are single edge argumenting pathes, we add all of them to M.

After these iterations, we achieved a matching M which is identical to the maximal matching provided in problem. Thus, through emulation, we have proven the Hopcroft-Karp algorithm is capable of producing the given matching.

2 Problem 2

This problem will inherit the labelling, `BFS` rule, and list M introduced by the above problem.

For $D, E, F \in U$, all of their conncted nodes $\in V$ are not free. Thus, stipulated by the Hopcroft-Karp algorithm, we will check for the free nodes $\in U$ for non-single-edged *argumenting paths*.

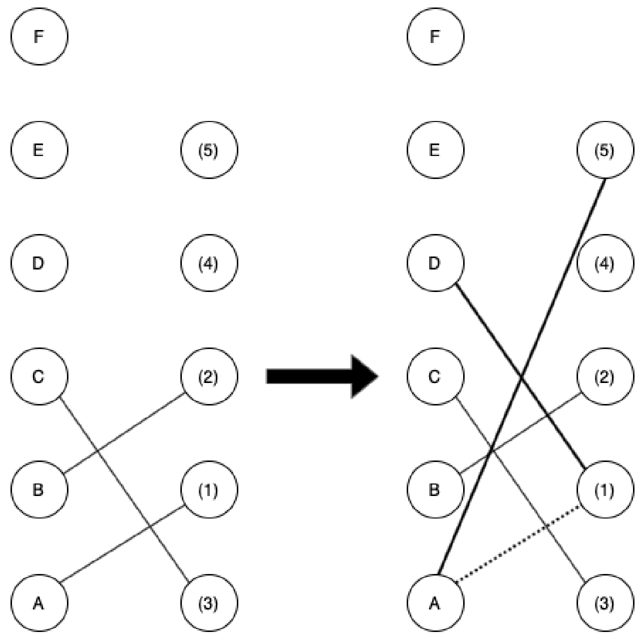

Figure 2: Obtaining a maximum matching of G from a given maximal matching

Starting with node D, as it has the lowerest alphabetical value of all free nodes $\in U$. It has two child nodes $\in V$, $\{(3), (1)\}$. We first check node (1) as it has a lower numerical value. Since we are looking for an *argumenting paths*, the only node we can proceed from (1) is A, as it is the only *matching node* (a node $\in M$) connected to (1). Then, the only node we can proceed from A is (5), as it is the only free node conncted to A.

Now, we do a `DFS` from (5) to D, we have discovered an *argumenting paths* of $< D, (1), A, (5) >$. Thus, we remove the *matching edge* $< A, (1) >$ from M, then add edge $< D, (1) >$ and $< A, (5) >$ to M.

Since there will be no more *argumenting paths* in the graph (as we can't find any more alternating path which starts and ends on free nodes). This $\{< C, (3) >, < B, (2) >, < D, (1) >, < A, (5) >\}$ matching, with a cardinality of 4, be the *maximum matching* for the graph.

3 Problem 3

Imaging a subgraph F of G, which is consisted by and only by edges in $M \sqcup N$ (edges that only $\in M$ or only $\in N$). Since $|M| > |N|$, we know that there must be $M \cap F > N \cap F$. Also for the ease of description, we initialize $M' = M$ and $N' = N$.

We also know that every node in F will either have degree of 1, where such node belongs and only belongs to M or N; or degree of 2, where such node is connected to an edge in M, and also conncted to an edge in N.

We first inspect edges in F. If there is an edge $E \in M$ from F that is a single edge (connected between two degree 1 nodes), we remove it from M' and add it to N'. This will make $|M'| = |M| - 1$ and $|N'| = N + 1$. We may tell that the new M' and N' are still matchings of G, due to the fact that for a single edge $\in M \sqcup N$, its two nodes are not connected to any edge of $M \cap N$, thus we won't have two edges of N' sharing a node of E.

If there is no single $E \in M$ from F, since $M \cap F > N \cap F$, there must be an alternating path P in F which starts and ends on edges of M. We inspect edges on this path: if an edge $EP \in P$ is also in M, we remove it from M' and add it to N'; vice versa, if an edge $EP \in P$ is also in N, we remove it from N' and add it to M'. Namely, we swapped the attribution of edges in P.

This maneuver will make $|M'| = |M| - 1$ and $|N'| = N + 1$, since path P starts and ends on edges of M, there must be one more M edge in P than N edges. Also the new M' and N' are still matchings of G, as the start- and end-nodes of path P are not connecting to any edges from $M \cap N$.

In both cases, we will have $M \cup N = M' \cup N'$ and $M \cap N = M' \cap N'$. Since we only swapped edge(s) in $M \sqcup N$ – which won't effect the intersection and union of the two matchings.