Algorithm Lab

Week 11: Solving Maximum Matching Problem

In a graph $G = (V, E)$, a match is an edge $e \in E$, and a matching is a set of matches $M \subseteq E$ that each vertex could be chosen at most one time, in other words, M is an independent edge set, thus $|V(M)| = 2|M|$. The maximum matching problem is to finding maximum independent edge set from a graph $G = (V, E)$.

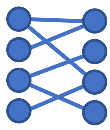
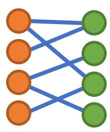
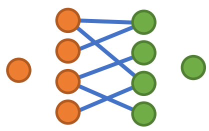
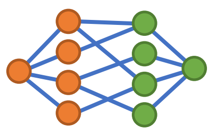
Instance: $G = (V, E)$

Result: Maximum matching $M \subseteq E$ *(or just* $|M|$*)*

If the instance is a bipartite graph, we say the problem is a bipartite matching problem.

Description

We can reduce a bipartite matching problem to a s-t maximum flow network.

1. Instance: $G = (V, E)$ 
2. Divide V to 2 groups V_s and V_t. 
3. Create vertices s and t. 
4. Let s connect to each $u \in V_s$ and each $v \in V_t$ connect to t. 
5. Set the capacity of all edges in this network to 1.
6. Compute the maximum s-t flow in the network by any maximum flow algorithm.

Questions

1 Describe the difference between original Ford-Fulkerson algorithm and Edmonds-Karp

algorithm in general flow network and in bipartite matching.

2 Brief about blossom algorithm.

3 Suppose edges between V_s and V_t are weighted, please design an algorithm for finding maximum matching with minimum cost.

4 Please compare time complexity between original flow algorithm and your modified version in Question 3.

5 Solve http://oj.csie.ndhu.edu.tw/problem/ALG11