CIS 575. Introduction to Algorithm Analysis
Material for April 24, 2024

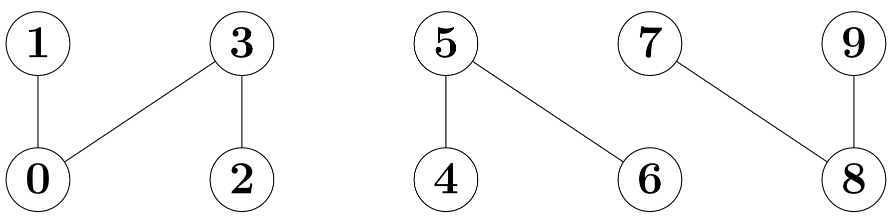
Bipartite Matching

Torben Amtoft

The topic of this note is covered in *Cormen*'s Section 24.3.

1 Matching in Bipartite Graphs

A **bipartite** graph is an undirected graph (V, E) where it is possible to partition V into two disjoint sets, V_1 and V_2, such that each edge in E is between a node in V_1 and a node in V_2. As an example, let us consider the graph



where the odd-numbered nodes are in V_1 and the even-numbered nodes are in V_2.

A **matching** M is a set of edges in E such that no node in V occurs more than once in M.

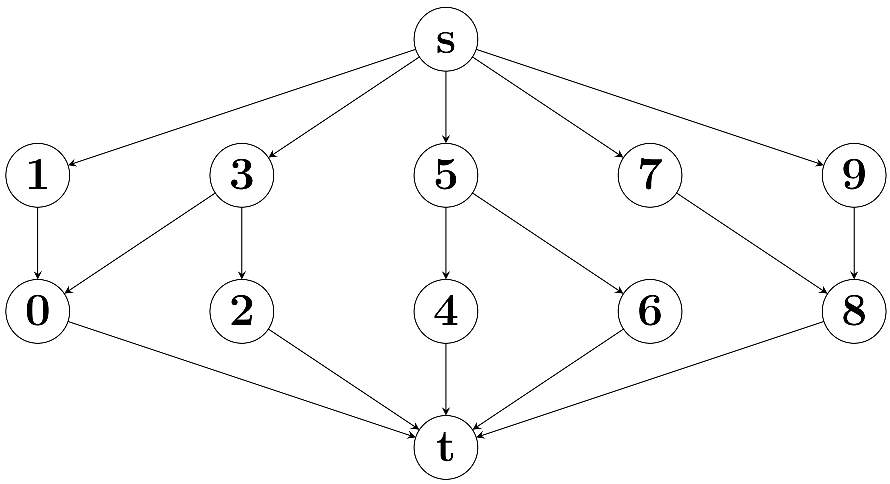
Our *goal* is to find a matching that is *maximal*, that is: no other matching has more edges.

For our example, a maximal matching will pair 1 with 0, 3 with 2, 5 with either 4 or 6, and 8 with either 7 or 9.

Maximal Matching as Network Flow Problem From a bipartite graph (V, E) with V the disjoint union of V_1 and V_2, we can construct a flow network (V', E') by letting

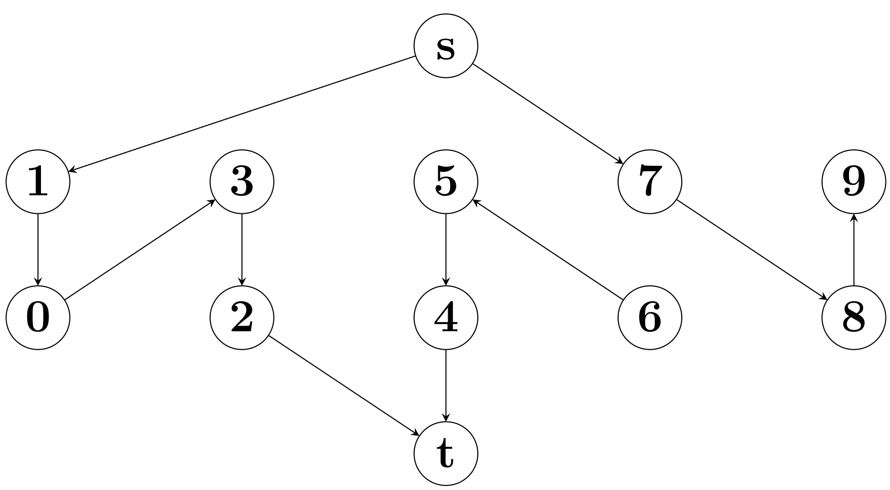
- $V' = V \cup \{s, t\}$ where the source s and the sink t are fresh nodes
- E' contain an edge from s to each node in V_1, and an edge from each node in V_2 to t
- E' contain an edge from $v_1 \in V_1$ to $v_2 \in V_2$ whenever E has an edge between v_1 and v_2
- all edges have (implicit) capacity 1.

Our example bipartite graph will yield the flow network

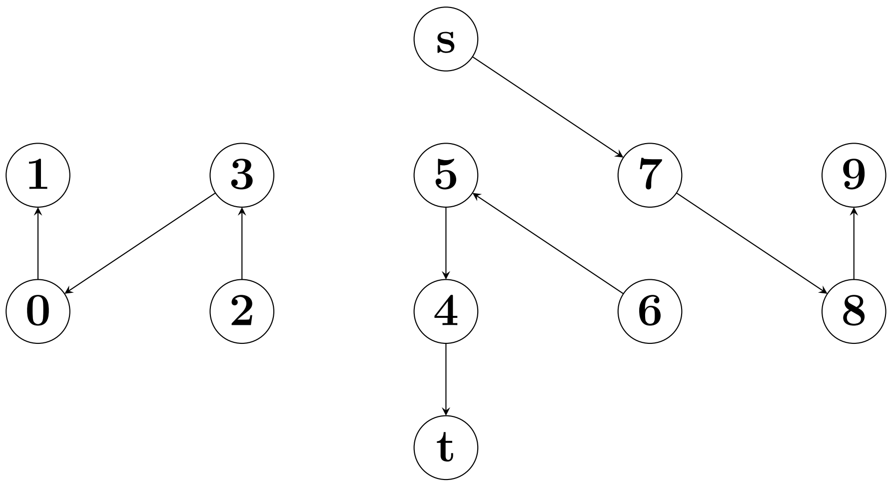


Observe that a bipartite graph has a matching with m edges iff the corresponding network has a flow with value m, where (v_1, v_2) belongs to the matching iff it has flow 1. For our example, the matching that pairs 0 with 3 and 7 with 8 corresponds to the flow, with value 2, that assigns 1 to the edges $(s, 3), (s, 7), (3, 0), (7, 8), (0, t), (8, t)$.

Maximal Matching by Ford-Fulkerson We can thus apply the Ford-Fulkerson method to construct a maximal matching. For our example, we may first choose augmenting paths $s30t$, $s56t$, and $s98t$, corresponding to the pairs $(3, 0), (5, 6), (9, 8)$. This yields the residual network



which has only one augmenting path: $s1032t$ that corresponds to removing the pair $(3, 0)$ and adding the pairs $(1, 0), (3, 2)$. We end up with the residual network



which has no augmenting path, and which enables us to see that one minimal cut is $\{s, 7, 8, 9\}$. As the Ford-Fulkerson method never decreases the flow of an edge from s or an edge into t, we see that a node once matched will remain matched, though perhaps to another node (as happened to node 0, and to node 3, in our example).