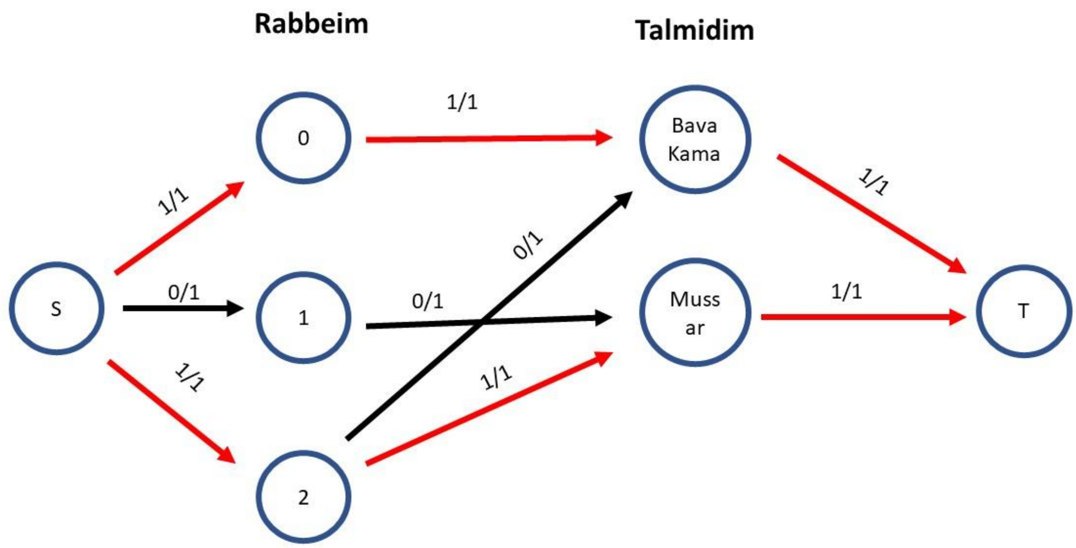
<u>Help From Rabbeim</u>

Jonathan Wenger

1. This problem is identical to the "bipartite matching" problem. In the bipartite matching problem, we are given a set of people L (referring to the Rabbeim) and a set of jobs R (referring to the students), and we want to have a matching that contains as many edges as possible. To have this problem be a network flow problem, we create a graph with a source vertex S, and S has an edge to each Rebbe vertex with an edge weight of 1. Additionally, each Talmid vertex has an edge to destination vertex T with an edge weight of 1. Finally, for every Talmid that a Rebbe is able to help, we connect an edge from all Rabbeim that can help to that talmid each with an edge weight of 1. After finishing to create the graph, we call the Ford-Fulkerson algorithm to find the maximum flow from S to T. If the maximum flow is less than the number of talmidim, then not every talmid is able to be helped, and then we return an empty map. If the maximum flow is equal to the number of talmidim, then every talmid can be helped, and we find which edges between Rabbeim vertices and Talmidim vertices have a flow of 1, and that's how we know which Rebbe is helping which Talmid.
2. This is taken from Slides 15-19 of the Applications of Network Flow slide deck. Because each Talmid vertex has exactly one edge to T, the conservation rule implies that each Talmid vertex uses at most one of its incoming edges. Similarly, because each Rebbe vertex has exactly one edge from S, the conservation rule implies that each Rebbe vertex uses at most one of its outgoing edges. This implies that each Rebbe vertex will be paired with at most one Talmid vertex, which, in other words, produces a matching. This also produces a maximum matching because the size of this matching must equal the maximum flow. If there was a better matching, then it wouldn't be a maximum flow. Additionally, based on the Flow Integrality Theorem, if all edge capacities are integers, then the max flow must also have an integer value, and Ford-Fulkerson will find a max flow such that f (u, v) is an integer for all edges (u, v). Because every edge in this graph has a capacity of 1, max-flow will either use the edge fully (sending 1 unit of flow), or not use it at all (sending 0 units of flow). Additionally, the amount of flow leaving S must be the same amount of flow entering T (the flow conservation constraint). Therefore, this shows that network flow actually solves this Help From Rabbeim problem.

3.



Rabbeim
Talmidim
0
1
2
S
Bava Kama
Muss ar
T
1/1
0/1
1/1
1/1
0/1
0/1
1/1
1/1
1/1