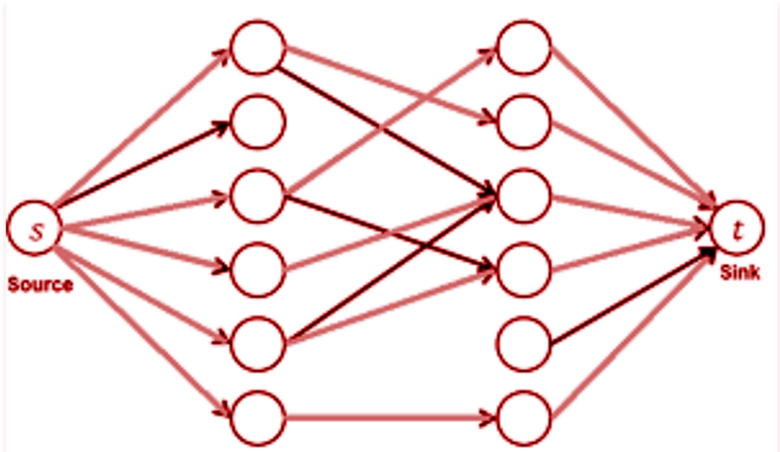
What is Maximum Bipartite Matching ?

We are given a bipartite graph **G** containing **V** vertices and **E** edges. Find the maximum matching, i.e., select as many edges as possible so that no selected edge shares a vertex with any other selected edge.

MaxFlow Solution

We can model this problem as a flow problem.
Here are the following steps.

1. Add a source and a sink vertex to the network.
2. Now Connect all vertices on one partition of the bipartite graph to the source and other to the sink.
3. Set the capacities of all the edges in the graph to 1. This will ensure few things:
    a. No vertex can be included in more than one edge due to edges with source/sink.
    b. The max flow in the graph will be equal to maximum matching in the original graph because size of matching M <= max(|A|, |B|) where A and B are two partitions of the graph. So maximizing flow in the graph is equivalent to maximizing the size of matching.
4. Finally we run any known flow algorithm like Edmond's Karp and return the max flow as the size of maximum matching.



The running time of the above solution is same as that of Edmond's Karp algorithm which is O(E*(V + E)) where E is the number of edges and V is the total number of vertices, since the max flow can be at most E.

However we will look at a better known and efficient algorithm for this problem which is known as Hopcroft-Karp.