1. The Ford-Fulkerson Algorithm runs in $O(C|E|)$ time where $C = \sum e \text{ out of } s \ c(e)$
   (a) Since the capacities remain integers, the amount of flow leaving $s$ (the value of the flow)
       increases by at least 1 on each iteration.
   (b) The algorithm therefore must terminate within $C$ iterations
   (c) Since depth-first search can find the paths in time $O(|E|)$ (if we have no isolated vertices)
       and computing the bottleneck and augment operations also can be done in time $O(|E|)$,
       the algorithm runs in $O(C|E|)$ time.

2. We will show correctness of Ford-Fulkerson using a structural bound involving “cuts” in $G$
   (a) A $s$-$t$ cut is a partition of $V$ into $A$ and $B$ such that $s \in A$ and $t \in B$
   (b) The capacity of the cut $(A, B)$ is the total capacity of edges crossing from $A$ to $B$
   (c) For any $s$-$t$ cut $(A, B)$, the value of a flow $f$ (from $s$ to $t$) is equal to the amount of flow
       leaving $A$ minus the amount of flow returning to $A$. Therefore, the value of any flow is
       at most the capacity of any cut.

3. Ford Fulkerson returns a flow with value equal to the capacity of a cut, and is therefore of
   maximum possible value (thus also “Max-Flow = Min-Cut”)
   (a) F-F terminates when there is no $s$-$t$ path using edges with positive residual capacity
   (b) If we look at the set of vertices reachable from $s$ using only edges of positive residual
       capacity $A^*$ when F-F terminates, the edges leaving $A^*$ must be saturated and there
       must not be any flow returning to $A^*$.
   (c) Therefore, the value of the flow is equal to the capacity of the corresponding cut.
   (d) Moreover, we can therefore find a minimum cut by finding this set $A^*$ using reachability
       in the residual graph – also in total time $O(C|E|)$ where $C$ is the total capacity of edges
       leaving $s$.

4. We will solve the Maximum Bipartite Matching problem by reducing it to (Integer) Maximum
   Flow, which is solved by F-F.
   (a) A bipartite graph is one that has a vertex set consisting of two disjoint sets $L$ and $R$
       such that all edges cross between $L$ and $R$. A matching is a set of edges that do not
       share any endpoints.
   (b) Maximum Bipartite Matching: Given a bipartite graph, find a matching of maximum
       size.