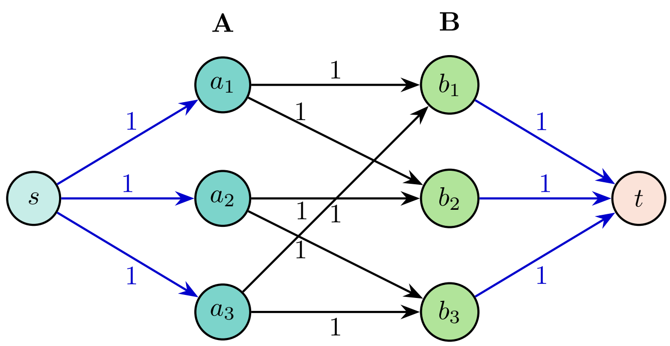
Minitest 6 – Solutions

1. For every 2-colorable graph G, we can use a flow problem to determine whether G has a perfect matching.

Answer: TRUE

Explanation. A 2-colorable graph is by definition a bipartite graph. Finding a maximum matching (and checking if it is perfect) in a bipartite graph $G = (A \cup B, E)$ can be directly modeled as a maximum flow problem by adding a source s with directed edges to all vertices in A, directing all original edges from A to B, and directing all vertices in B to a sink t, with all capacities set to 1 (details in Script on page 183).



2. Let $N = (V, A, c, s, t)$ be a network. If c has only integer edge weights, then every maximum flow in N is integral.

Answer: FALSE

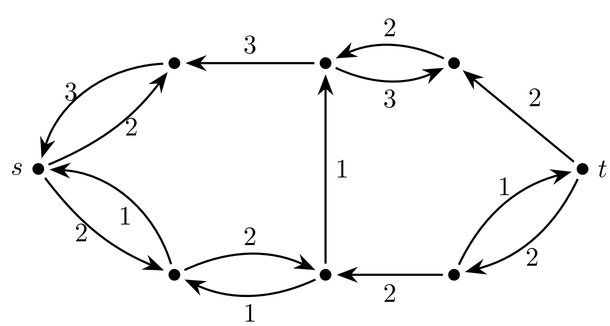
Explanation. According to Satz 3.12. in Script an integral maximum flow must **exist** (under the assumptions). However, it is not guaranteed that **every** maximum flow is integral. Consider, for example, the network above (in statement 1). Despite integral capacities, we could also create a maximum flow by having flow of 0.5 on all edges from A to B.

3. Let $N = (V, A, c, s, t)$ be a network without oppositely directed edges, and let f be a flow such that $\text{val}(f) > 0$. Let N^f be the residual network. Then N^f contains a directed path from t to s.

Answer: TRUE

Explanation. Since $\text{val}(f) > 0$, there is positive flow sent from s to t. Every unit of forward flow along an edge (u, v) creates a backward residual capacity along the reverse edge (v, u) in the residual network N^f. By tracing these reverse residual paths backwards along the tracks of the active flow, we are guaranteed to find a directed path from the sink t back to the source s [Definition 3.10. "Restnetzwerk"].

4. Let $N = (V, A, c, s, t)$ be a network without oppositely directed edges, and let f be a flow. The residual network N^f is given as follows:



Is the flow maximal?

Answer: TRUE

Explanation. t is not reachable from s in N^f. Hence, according to Satz 3.11. in Script f is a maximum flow.

5. Let $N = (V, A, c, s, t)$ be a network without oppositely directed edges. Suppose that the capacity of every edge is at most U. Then the Ford-Fulkerson algorithm computes a maximum flow in time $O(mnU)$.

Answer: FALSE

Explanation. The assumption that the capacities are integral is missing. Hence, Satz 3.12. in Script cannot be applied. Ford-Fulkerson might not terminate at all when working with irrational capacities [Script page 181].

6. Let $N = (V, A, c, s, t)$ be a network, let f be a flow in N, and let (S, T) be an s-t cut in N. Then val$(f) \geq$ cap(S, T).

Answer: FALSE

Explanation. One could choose any non maximum flow as a counterexample [Lemma 3.8. and Satz 3.9.].

7. Let f be a maximum flow in a network $N = (V, A, c, s, t)$. Then there are no two nodes $u, v \in V$ such that both the edge (u, v) and the edge (v, u) occur in the residual network N^f.

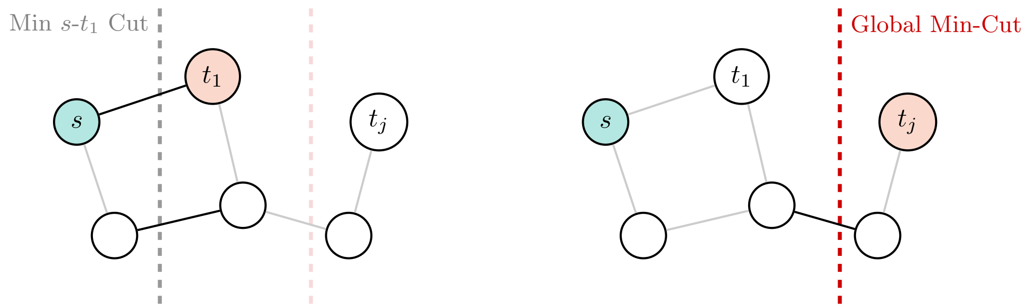
Answer: FALSE

Explanation. It is entirely possible for both forward and backward edges to exist in the residual network N^f. For instance, if an edge (u, v) has capacity $c(u, v) = 10$ and carries flow $f(u, v) = 4$, the residual network will contain a forward edge (u, v) with residual capacity $10 - 4 = 6$, and a backward edge (v, u) with residual capacity 4 [Definition 3.10. "Restnetzwerk"].

8. The MIN-CUT problem can be solved by fixing one vertex s and computing minimum s-t cuts for all $t \in V \setminus \{s\}$.

Answer: TRUE

Explanation. [Script page 192] Inelegant illustration:



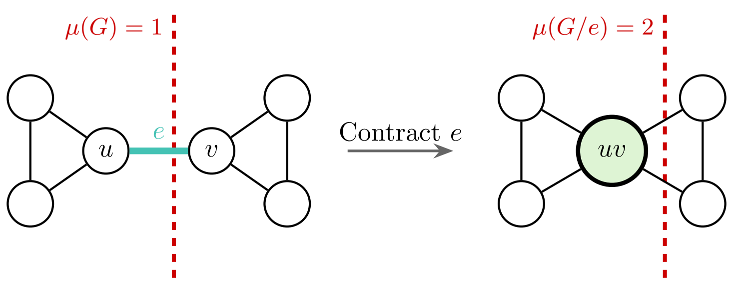
Iteration 1: Trying with t_1
s, t_1 are on **different** sides of cut.
Cut size = 2
(we first need to try with all
other t's to see if this is $\mu(\mathbf{G})$)

Iteration j: Trying with t_j
s, t_1 are on **different** sides of cut.
Cut size = 1 = $\mu(\mathbf{G})$

9. If e is an edge of G, then always $\mu(G/e) \leq \mu(G)$.

Answer: FALSE

Explanation. Here $\mu(G)$ denotes the size of the minimum cut (edge connectivity) of the graph. **Counterexample:**



Original Graph G

Contracted Graph G/e

10. If there exists a minimum cut C of G with $e \notin C$, then $\mu(G/e) = \mu(G)$.

Answer: TRUE

Explanation. If the edge e is not part of the minimum cut C, then both endpoints of e lie entirely on the same side of the cut partition (A, B) (either entirely in A or entirely in B). Contracting e merges these two vertices inside their respective partition sub-set without destroying or altering any of the edges that cross between A and B.