Required Problems

1. Scalar Flow Product  
   [10 Points] (Based on CLRS 26.1-7)  
   Let \( f \) be a flow in a network, and let \( \alpha \) be a real number. The \textit{scalar flow product}, denoted by \( \alpha f \), is a function from \( V \times V \) to \( \mathbb{R} \) defined by  
   \[
   (\alpha f)(u,v) = \alpha \cdot f(u,v).
   \]
   Prove that the flows in a network form a \textit{convex set}. That is, show that if \( f_1 \) and \( f_2 \) are flows, then so is \( \alpha f_1 + (1 - \alpha)f_2 \) for all \( \alpha \) in the range \( 0 \leq \alpha \leq 1 \).

2. (Based on CLRS 26.1-9)  
   [10 Points]  
   Professor Adam has two children who, unfortunately, dislike each other. The problem is so severe that not only they refuse to walk to school together, but in fact each one refuses to walk on any block that the other child has stepped on that day. The children have no problem with their paths crossing at a corner. Fortunately both the professor’s house and the school are on corners, but beyond that he is not sure if it is going to be possible to send both of his children to the same school. The professor has a map of his town. Show how to formulate the problem of determining if both his children can go to the same school as a maximum-flow problem.
3. (Based on CLRS 26.2-8 and 26.2-10)  
[10 Points]

(a) [5 Points] Show that a maximum flow in a network \( G = (V, E) \) can always be found by a sequence of at most \(|E|\) augmenting paths. [Hint: Determine the paths after finding the maximum flow.]

(b) [5 Points] Suppose that a flow network \( G = (V, E) \) has symmetric edges, that is, \((u, v) \in E\) if and only \((v, u) \in E\). Show that the Edmonds-Karp algorithm terminates after at most \(|V||E|/4\) iterations. [Hint: For any edge \((u,v)\), consider how both \(\delta(s, u)\) and \(\delta(v, t)\) change between times at which \((u,v)\) is critical.]

4. **Edge Connectivity**  
[10 Points] (Based on CLRS 26.2-9)

The **edge connectivity** of an undirected graph is the minimum number \(k\) of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how the edge connectivity of an undirected graph \( G = (V, E) \) can be determined by running a maximum-flow algorithm on at most \(|V|\) flow networks, each having \(O(V)\) vertices and \(O(E)\) edges.

5. **Perfect Matching**  
[20 Points] (Based on CLRS 26.3-4 and 26.3-5)

(a) [10 Points] A **perfect matching** is a matching in which every vertex is matched. Let \( G = (V, E) \) be an undirected bipartite graph with vertex partition \( V = L \cup R \), where \(|L| = |R|\). For any \(X \subseteq V\), define the **neighborhood** of \(X\) as

\[
N(X) = \left\{ y \in V \mid (x, y) \in E \text{ for some } x \in X \right\},
\]

that is, the set of vertices adjacent to some member of \(X\). Prove Hall’s theorem: there exists a perfect matching in \(G\) if and only if \(|A| \leq |N(A)|\) for every subset \(A \subseteq L\).

(b) [10 Points] We say that a bipartite graph \( G = (V, E) \), where \( V = L \cup R \), is **\(d\)-regular** if every vertex \(v \in V\) has degree exactly \(d\). Every \(d\)-regular bipartite graph has \(|L| = |R|\). Prove that every \(d\)-regular bipartite graph has a matching of cardinality \(|L|\) by arguing that a minimum cut of the corresponding flow network has capacity \(|L|\).

6. **Maximum Flow By Scaling**  
[20 Points] (Based on CLRS 26-5)

Let \( G = (V, E) \) be a flow network with source \(s\), sink \(t\), and an integer capacity \(c(u, v)\) on each edge \((u, v) \in E\). Let \( C = \max_{(u, v) \in E} c(u, v) \).

(a) [2 Points] Argue that a minimum cut of \(G\) has capacity at most \(C|E|\).
(b) [5 Points] For a given number K, show that an augmenting path of capacity at least K can be found in O(E) time, if such a path exists.

The following modification of Ford-Fulkerson-Method can be used to compute a maximum flow in G.

```
Max-Flow-By-Scaling(G, s, t)
1  C ← max_{(u,v) ∈ E} c(u,v)
2  initialize flow f to 0
3  K ← 2^\lfloor \log C \rfloor
4  while K ≥ 1 do {
5      while (there exists an augmenting path p of capacity at least K) do {
6          augment flow f along p
7      }
8      K ← K/2
9  }
10  return f
```

(c) [3 Points] Argue that Max-Flow-By-Scaling returns a maximum flow.

(d) [4 Points] Show that the capacity of a minimum cut of the residual graph G_f is at most 2K|E| each time line 4 is executed.

(e) [4 Points] Argue that the inner while loop of lines 5-6 is executed O(E) times for each value of K.

(f) [2 Points] Conclude that Max-Flow-By-Scaling can be implemented so that it runs in O(E^2 log C) time.

7. The Hopcroft-Karp Bipartite Matching Algorithm

[20 Points] (Based on CLRS 26-7)

In this problem, we describe a faster algorithm, due to Hopcroft and Karp, for finding a maximum matching in a bipartite graph. The algorithm runs in O(√VE) time. Given an undirected, bipartite graph G = (V,E), where V = L ∪ R and all edges have exactly one endpoint in L, let M be a matching in G. We say that a simple path P in G is an augmenting path with respect to M if it starts at an unmatched vertex in L, ends at an unmatched vertex in R, and its edges belong alternately to M and E − M. (This definition of an augmenting path is related to, but different from, an augmenting path in a flow network.) In this problem, we treat a path as a sequence of edges, rather than as a sequence of vertices. A shortest augmenting path with respect to a matching M is an augmenting path with a minimum number of edges.

Given two sets A and B, the symmetric difference A ⊕ B is defined as (A − B) ∪ (B − A), that is, the elements that are in exactly one of the two sets.
(a) **[4 Points]** Show that if $M$ is a matching and $P$ is an augmenting path with respect to $M$, then the symmetric difference $M \oplus P$ is a matching and $|M \oplus P| = |M| + 1$. Show that if $P_1, P_2, ..., P_k$ are vertex-disjoint augmenting paths with respect to $M$, then the symmetric difference $M \oplus (P_1 \cup P_2 \cup ... \cup P_k)$ is a matching with cardinality $|M| + k$.

The general structure of our algorithm is the following:

```
HOPCROFT-KARP(G)
1    M ← ∅
2    repeat
3        let $\mathcal{P} \leftarrow \{P_1, P_2, ..., P_k\}$ be a maximum set of
4            vertex-disjoint shortest augmenting paths
5            with respect to $M$
6        M ← $M \oplus (P_1 \cup P_2 \cup ... \cup P_k)$
7    until $\mathcal{P} = ∅$
8    return $M$
```

The remainder of this problem asks you to analyze the number of iterations in the algorithm (that is, the number of iterations in the repeat loop) and to describe an implementation of line 3.

(b) **[4 Points]** Given two matchings $M$ and $M^*$ in $G$, show that every vertex in the graph $G' = (V, M \oplus M^*)$ has degree at most 2. Conclude that $G'$ is a disjoint union of simple paths or cycles. Argue that edges in each such simple path or cycle belong alternately to $M$ or $M^*$. Prove that if $|M| \leq |M^*|$, then $M \oplus M^*$ contains at least $|M^*| - |M|$ vertex-disjoint augmenting paths with respect to $M$.

Let $l$ be the length of a shortest augmenting path with respect to a matching $M$, and let $P_1, P_2, ..., P_k$ be a maximum set of vertex-disjoint augmenting paths of length $l$ with respect to $M$. Let $M' = M \oplus (P_1 \cup P_2 \cup ... \cup P_k)$, and suppose that $P$ is a shortest augmenting path with respect to $M'$.

(c) **[2 Points]** Show that if $P$ is vertex-disjoint from $P_1, P_2, ..., P_k$, then $P$ has more than $l$ edges.

(d) **[2 Points]** Now suppose $P$ is not vertex-disjoint from $P_1, P_2, ..., P_k$. Let $A$ be the set of edges $(M \oplus M') \cap P$. Show that $A = (P_1 \cup P_2 \cup ... \cup P_k) \cap P$ and that $|A| \geq (k + 1)l$. Conclude that $P$ has more than $l$ edges.

(e) **[2 Points]** Prove that if a shortest augmenting path for $M$ has length $l$, the size of the maximum matching is at most $|M| + |V|/l$.

(f) **[2 Points]** Show that the number of repeat loop iterations in the algorithm is at most $2\sqrt{V}$. [Hint: By how much can $M$ grow after iteration number $\sqrt{V}$?]

(g) **[4 Points]** Give an algorithm that runs in $O(E)$ time to find a maximum set of vertex-disjoint shortest augmenting paths $P_1, P_2, ..., P_k$ for a given matching $M$. Conclude that the total running time of HOPCROFT-KARP is $O(\sqrt{VE})$. 

4