1. Computability, Complexity and Algorithms

Bottleneck edges in a flow network:
Consider a flow network on a directed graph $G = (V, E)$ with capacities $c_e > 0$ for $e \in E$. An edge $e \in E$ is called a bottleneck edge if increasing the capacity $c_e$ increases the size of the maximum flow.

Given a flow network $G = (V, E)$ and a maximum flow $f^*$, give an algorithm to identify all bottleneck edges. Do as fast in $O(\cdot)$ as possible. Justify correctness of your algorithm. You can assume basic operations (comparison, addition, subtraction, multiplication, and division) on two numbers take constant time.

2. Analysis of Algorithms

All-pairs shortest paths (APSP) and Min-Sum Products. Suppose $W$ is the adjacency matrix for $G$ a simple undirected graph with no self-loops and no negative edge weights, and $W^*$ is the reachability matrix ($w^*_{ij} = 1$ if there exists a path from $i$ to $j$).

- Suppose operations are boolean (addition is OR, multiplication is AND). Suppose
  
  \[
  W = \begin{bmatrix}
  A & B \\
  C & D
  \end{bmatrix}
  \]

  Then show that
  
  \[
  W^* = \begin{bmatrix}
  E & F \\
  G & H
  \end{bmatrix} = \begin{bmatrix}
  (A \lor BD^*C)^* & EBD^* \\
  D^*CE & D^* \lor GBD^*
  \end{bmatrix}
  \]

  Observe that $F, G$ use $E$ in their definition, etc., so the calculations have to be done in the correct order. *Hint:* Consider $G$ as partitioned into two subcomponents $V = V_1 \cup V_2$.

- Now suppose $W_{ij}$ is the weight of the edge $(i, j)$. Moreover, now assume that matrix products are min-sum products (that is, addition is replaced by min and product by sum), and $A \lor B$ is the element-wise minimum of matrices $A$ and $B$. If $W^*_{ij}$ now denotes the shortest-path distance from $i$ to $j$, show that $W^*$ is computed by the same relation as in the previous part. You may be brief, 2-3 sentences suffices if your previous answer was thorough.

- Using this idea, show that
  
  \[
  \text{APSP}(n) \leq 2\text{APSP}(n/2) + 6\text{MSP}(n/2) + O(n^2),
  \]

  where APSP($n$) denotes the worst-case running time of computing APSP on an $n$-vertex input graph, and MSP($n$) denotes the worst-case running time of computing the min-sum product of two $n \times n$ matrices. Assume that arithmetic operations can be carried out in constant time.

  In turn, show that $\text{APSP}(n) = \tilde{O}(\text{MSP}(n) + n^2)$. *Hint:* We know that MSP is superlinear, even superquadratic, in its runtime, simply since it needs to read its two input matrices.
3. Theory of Linear Inequalities

Let \( P = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \subseteq [0,1]^n \) be a polytope with 0/1 vertices. It is well known that the diameter of any 0/1 polytope is at most \( n \). Here we consider a stronger notion of diameter where the sequence of vertices has to be non-decreasing in value with respect to a given objective \( c \in \mathbb{Z}^n \): For any two vertices \( x, y \in P \) with \( cy = \max_{z \in P} cz \) find the shortest path of adjacent vertices \( x_1, \ldots, x_l \) with \( x = x_1 \) and \( y = x_l \) so that \( cx = cx_1 \leq \cdots \leq cx_l = cy \). The monotone diameter for an objective \( c \) is the maximum length over all such vertex pairs.

Prove that the monotone diameter is at most \( O(n \log C) \), where \( C = \max_i |c_i| \) (6 points). Can you also show that in this case the monotone diameter is at most \( n \) irrespective of the objective \( c \)? (4 points)

4. Combinatorial Optimization

Let \( \mathcal{M} = (U, \mathcal{I}) \) be a matroid and \( w : U \to \mathbb{R} \) be a weight function.

1. Given any two bases \( B \) and \( B' \), show that there exists a sequence of bases \( B_0, B_1, \ldots, B_k \) with the following properties.

   (a) \( B_0 = B \) and \( B_k = B' \).

   (b) \( B_i \subseteq B \cup B' \) for each \( 0 \leq i \leq k \).

   (c) \( |B_i \Delta B_{i+1}| = 2 \) for each \( 0 \leq i \leq k - 1 \).

2. Suppose \( B' \) is a maximum weight basis under weight function \( w \). Show that we can additionally ensure that \( w(B_{i+1}) \geq w(B_i) \) for each \( 0 \leq i \leq k - 1 \).

5. Graph Theory

Let \( G \) be a 2-connected graph and let \( s \in V(G) \). Prove that \( G \) has two spanning trees \( T_1, T_2 \) such that for every vertex \( v \in V(G) \) the two paths between \( v \) and \( s \) in \( T_1 \) and \( T_2 \) are internally disjoint.

6. Probabilistic methods

Suppose that we throw \( m \) balls into \( n \) bins independently and uniformly at random (initially all bins are empty, of course).

(A) Prove that \( m^*(n) = n \log n \) is a threshold function for the property ‘there exists an empty bin’, i.e.,

\[
\Pr(\text{there exists an empty bin}) \rightarrow \begin{cases} 
1 & m \ll n \log n, \\
0 & m \gg n \log n.
\end{cases}
\]
(B) Make an educated guess what the threshold function for the property ‘there exists a bin with at most one ball’ is. Prove the corresponding 0-statement (no proof of the corresponding 1-statement expected).

**Hint:** Recall that \( 1 - x = e^{-x + O(x^2)} \) as \( x \to 0 \).

7. **Algebra**

Suppose \( p \) and \( q \) are odd primes and \( p < q \). Let \( G \) be a finite group of order \( p^3q \). Prove that \( G \) has a normal Sylow subgroup.