COL 751 : Practice Sheet - 4

1. Strong-edge-connectivity
Prove or disprove: Every k-strongly-edge-connected digraph G is $2k$-edge-connected after ignoring edge directions.

2. Bounded Edge-Connectivity Preserver
Let $G = (V, E)$ be an undirected graph and $k \geqslant 1$ be an integer. Prove that there exists a subgraph $H = (V, E_H)$ of G with at most $k(n-1)$ edges such that for each $x, y \in V$ following holds:

$$\text{MAX-FLOW}(x, y, H) \;\geqslant\; \min\big\{k, \text{MAX-FLOW}(x, y, G)\big\}.$$

3. Bounded Flow Preserver
Let $G = (V, E)$ be a digraph with a source $s \in V$ and $k \geqslant 1$ be an integer. Prove that there exists a subgraph $H = (V, E_H)$ of G such that for each $x \in V$ following holds:

- in-degree$(x, H) \leqslant k$,
- $\text{MAX-FLOW}(s, x, H) \;\geqslant\; \min\big\{k, \text{MAX-FLOW}(s, x, G)\big\}$.

4. Disjoint trees in Streaming Model
Let G be a $2k$-edge-connected graph. Design an algorithm to obtain a single-pass *streaming algorithm* to compute k spanning trees $T_1, \ldots, T_k$ (i.e. trees with n vertices) that are edge-disjoint subgraphs of G in $O(nk^2 \log n)$ working space.

5. Augmenting a graph
Let $G = (V, E)$ be an undirected graph on n vertices $v_1, \ldots, v_n$, and $d_1, \ldots, d_n$ be collection of n positive degree constraints. Design an algorithm to compute a set $E^* \subseteq V \times V$ (if it exists) such that the following holds for augmented graph $G^* = (V, E \cup E^*)$:

i For $i \in [1, n]$, degree of v_i in G^* is at most d_i.

ii The edge connectivity of G^* is at least k.

Hint: Use Lovasz's Splitting Off Theorem to first add edges to an auxiliary vertex v_0 not in V.

6. Gomory-Hu Trees
(a) Let $T = (V, E_T)$ be a Gomory-Hu tree for a connected graph $G = (V, E)$. Prove that for any pair of vertices $s, t \in V$ and any (s, t) cut (X, X^c) in G, there is an edge $(a, b) \in E_T$ such that $a \in X$, $b \in X^c$, and (a, b) lies on the path between s and t in tree T.

(b) The Steiner k-cut problem is defined as follows. Given an edge-weighted undirected graph $G = (V, E)$, a subset of vertices $X \subseteq V$ called terminals, and an integer $k \leqslant |X|$, find a minimum weight set of edges whose removal results in k dis-connected components, each of which contains at least one terminal.

A natural greedy algorithm for the Steiner k-cut problem is as follows. Iteratively, pick the smallest weight edge in T separating a pair of terminals that are not already separated, until k components, each of which contains a terminal, are generated. It is easy to see that we pick $k-1$ edges in T. We take the union of the cuts associated with these edges and this is our solution for the Steiner k-cut problem in G. Prove that the cost of this solution is at most $(2-2/k)$ times that of the optimal solution.

7. Global Min-Cut

Obtain a bound on the success probability of the following algorithm for computing a global minimum-cut in an undirected unweighted graph G.

Algorithm 1 Compute-Min-Cut(G)

Initialize H_1, H_2 as G.
while $(|V(H_1)|, |V(H_2)| > n/2)$ **do**
  Pick a uniformly random edge e_1 in H_1, and contract it.
  Pick a uniformly random edge e_2 in H_2, and contract it.
$(A_1, B_1) =$ Compute-Min-Cut(H_1).
$(A_2, B_2) =$ Compute-Min-Cut(H_2).
Return the smaller of the two cuts (A_1, B_1) and (A_2, B_2).

8. Reporting all Global Min-Cuts

Design an algorithm that given any n vertex undirected graph $G = (V, E)$ computes all global min-cuts of G in $O(n^4 \log n)$ time, with high probability. Can a similar polynomial time bound be achieved for reporting all (s,t)-minimum-cuts, for a source s and destination t in G?

9. (s,t)-Mininum-Cuts

Design an algorithm that given any n vertex, m edges undirected graph G with source s and destination t determines if there are at least three distinct (s,t)-minimum-cuts in G in $O(mn)$ time.