PROBLEM 1. *T/F*

(a) Asymptotics: $(\log n)^{\log n} = \Omega(\log \log n)$

(b) For any 2 functions f and g, either $f = O(g)$ or $g = O(f)$.

(c) If all of the edge capacities in a graph are an integer multiple of 5, then the value of the maximum flow from source to sink will be a multiple of 5.

(d) In a directed graph $G = (V, E)$ with capacities c, source s and sink t, the minimum $s - t$ cut is always unique.

(e) In a connected weighted graph with distinct positive weights, the edge with maximum weight is never in the minimum spanning tree.

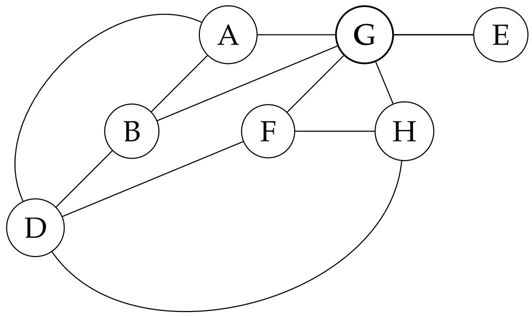
(f) Given n-bit numbers, A,B, their product can be calculated in $\Theta(n^{\log_2 3})$ time.

(g) **NP** stands for Not Polynomial Time.

(h) If there is a polynomial-time algorithm for any **NP**-Complete problem, then every problem in **NP** has a polynomial-time algorithm.

problem 2. *BFS*

Give the BFS traversal for the following graph G, starting at node A. Fill in the edges *with directions*. If there are multiple possible trees, any correct tree suffices.



problem 3. *VC to Clique*

(a) Define Vertex Cover for a graph $G = (V, E)$.

(b) Define k-Clique for G.

(c) Explain why: For any graph $G = (V, E)$ and subset $V' \subseteq V$, the following statements are equivalent:

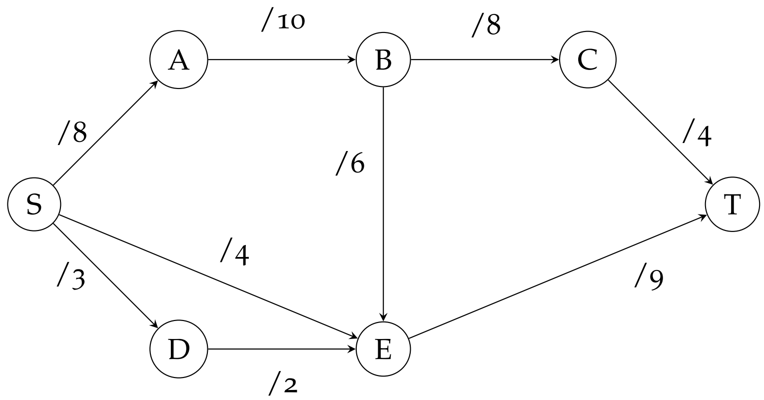
(a) V' is a vertex cover for G.

(b) $V \setminus V'$ is a clique in the complement G^c of G, where $G^c = (V, E^c)$ with $E^c = \{\{u, v\} : u, v \in V \text{ and } \{u, v\} \notin E\}$.

(d) Give a polynomial-time reduction Vertex Cover $\leq_p$ Clique.

problem 4. *Computing Max Flow*

Using the EK algorithm, calculate the maximum flow through the graph shown below. List each augmenting path and its bottleneck capacity.



PROBLEM 5. *Network Flow Reduction*

You need to assign TAs to questions to grade. There are n types of questions to grade, with t_i questions of type i, for $1 \le i \le n$. There are m TAs, where TA j has the time to grade at most c_j questions, and is capable of only grading questions of types drawn from a subset $S_j \subseteq \{1, 2, \ldots, n\}$ of types.

Give a polynomial-time algorithm to determine if it is possible to assign TAs to the questions so that all the questions are graded. If it is possible to do so, your algorithm should also return an assignment that indicates how many questions of each type are graded by each TA. There is no need to prove the correctness of the algorithm or state a running time for your algorithm.

problem 6. *VEB*

Draw a VEB queue universe $U = 256$, base case $U = 4$, containing 61,64,65.

problem 7. *Smallest Dominating Subset*

a Given a set of positive integers S, describe a greedy algorithm that runs in $O(n \log n)$ to find the smallest subset A of S such that the sum of elements in A is more than the sum of the remaining elements.

b Prove that your algorithm is optimal using the exchange argument and show that it runs in $O(n \log n)$ as required.

c Describe how you can modify your algorithm to run in $O(n + k \log n)$ where k is the size of the optimal subset A. For this, treat k as a parameter, not a constant.