Exam

Remarks: All the graphs here are without self loops and parallel or anti-parallel edges. A network is a directed graph with source $s$ and sink $t$ and capacity $c_e > 0$ on every edge $e$. In all the algorithms, always explain their correctness and analyze their complexity. The complexity should be as small as possible. A correct algorithm with large complexity, may not get full credit. The number of vertices is denoted by $n$, and the number of edges by $m$.

Choose 5 out of the next 6 questions.

**Question 1:** On all the next questions answer true or false and explain:

1. If there exists at least one $s$ to $t$ path in the network, then there exists a single edge $e$ so that if we increase the capacity of $e$ the max flow will increase
2. A perfect matching is a matching that contains all the vertices. True or false: Every tree has at most one perfect matching,
3. True or false: a minimum weight shortest path is still a minimum weight shortest path if cost 1 is added to every edge
4. If the min-cut algorithm of Karger returns a cut of capacity $\tau$ then with high probability the minimum cut of the network has value $\tau$

**Question 2:** A bipartite graph is $d$-regular if and only if the degree of every node is exactly $d$. Show that a $d$-regular bipartite graph always has a perfect matching (a matching of size $n/2$ that includes all vertices).

**Question 3:** A network with capacity on the nodes is a directed graph with source $s$, sink $t$, capacities over the edges and capacity $c(v)$ for every $v \in V$. It is required that aside from the usual conditions, at most $c(v)$ flow units enter $v$ along the edges of $v$. The goal is to maximize the flow entering $t$ ($t$ and $s$ have no capacities). Give an algorithm that solves the problem

**Question 4:** Show that the distance from $s$ to a vertex $v$ can not decrease in the residual graph during a run of the Ford and Fulkerson algorithm (the paths chosen are arbitrary)

**Question 5:** For a tree $T$ with weights on edges, let $\max(T) = \max_{e \in E} c(e)$. Give an algorithm that finds a tree with minimum $\max(T)$.

**Question 6:** Say that instead of a single source $s$ there are several sources $s_1, s_2, ....$ Thus, flow can leave any $s_i$ without any entering flow (according to capacity constrains). The goal is still to respect the $f(e) \leq c(e)$ constrains and to maximize the amount of flow entering $t$. Give an algorithm for this problem