Exercise 1. (3 + 6 points)
In this exercise we prove the invariants of the Duan Mehlhorn Alorithm. Prove that throughout all the iterations of the algoritphm.

(a) \(\sum_{i \in [n]} r_f(b_i) \leq n\).

(b) The maximum price of any good is less than \((nU)^{n-1}\).

Exercise 2. (total: 1+4+4+2 points)
In this exercise, we shall see that the Duan Mehlhorn Algorithm runs faster for some input instances. We define the Utility Graph \(U_G\) to be a bipartite graph defined on the vertices \(B \cup G\) and there are edges \((b_i, g_j)\) with \(b_i \in B\) and \(g_j \in G\) iff \(v_{ij} > 0\). Let \(D\) represent any arbitrary cycle in \(U_G\). Prove that,

(a) The Cycle is even. (the number of edges in the cycle is even).

Now Color the edges of \(D\) with two colors (red and blue say), such that no two adjacent edges have identical colors. Let us call the set of red colored edges as \(D_r\) and the set of blue colored edges as \(D_b\). We are also given that \(\prod_{e \in D_r} v_e \neq \prod_{e \in D_b} u_e\). (this holds for any arbitrary cycle). Show that

(b) The Equality network \(N_p\) is acyclic in every iteration of the algorithm.

(c) The maximum flow in \(N_p\) can be computed with \(O(n)\) arithmetic operations in any arbitrary iteration of the algorithm.

(d) The Duan Mehlhorn algorithm converges to equilibrium in \(O(n^7 \cdot \log(nU))\) arithmetic operations.

Exercise 3. (total: 9+1 points)
In this exercise we shall devise an algorithm for the generalized Linear Arrow Debreu Markets. We formally state the generalized Arrow-Debreau model \(AD_G(n, m)\) first. We are given a set \(B\) of \(n\) rational agents and \(G\) of \(m\) divisible goods. We will refer to an individual agent (or good) by \(b_i\) for \(i \in [n]\) (or \(g_j\) for \(j \in [m]\)). Additionally there are two functions \(W, U : B \times G \rightarrow \mathbb{R}\), where \(W(b_i, g_j)\) quantifies the amount of \(g_j\) owned by \(b_i\) and \(U(b_i, g_j)\) indicates the Utility \(b_i\) derives from
one unit of the good $g_j$. Our goal is to determine a non-negative price vector $p \in \mathbb{R}^{m \times 1}$ and non-negative money flows $f : B \times G \rightarrow \mathbb{R}$ (where $f(b_i, g_j)$ indicates the price of the amount of $g_j$ consumed by $b_i$) such that,

- $f(b_i, g_j) > 0$ only if $\frac{U(b_i, g_l)}{p_l} \geq \frac{U(b_i, g_k)}{p_k} \quad \forall k \in [m]$ (This conditions result from linearity of the utility functions and rationality of buyers).

- $\sum_{i \in [n]} f(b_i, g_j) = \sum_{i \in [n]} W(b_i, g_j) \cdot p_j \quad \forall j \in [m]$ (All goods are completely sold).

- $\sum_{j \in [m]} f(b_i, g_j) = \sum_{j \in [m]} W(b_i, g_j) \cdot p_j \quad \forall i \in [n]$ (All the buyers invest their budgets completely).

In the lecture we discussed a special variant of the above stated problem where every agent owns only all of one good (We call these family of instances $AD_S(n)$). We first show that the general case reduces to the special case. We first create $O(n \times m)$ many agents and goods (we refer to them as $b_{ij}$ and $g_{ij}$) such that if $W_{ij} > 0$, we have one unit of good $g_{ij}$ and this is owned by agent $b_{ij}$. Similarly, if $W_{ij} = 0$, then there is no good $g_{ij}$ and no agent $b_{ij}$. We interpret $g_{ij}$ as a copy of good $g_j$ and $b_{ij}$ as a copy of agent $b_i$. Hereafter we define $U(b_{ij}, g_{lk}) = W(b_l, g_k) \cdot U(b_i, g_k)$.

Let $p$ and $f$ be the market clearing price vector and the corresponding money flow for the above instance of $AD_S(n)$. Then show that,

(a) $\frac{p_{lk}}{W(b_l, g_k)}$ does not depend on $l$, but only on $k$. Let $\hat{p}_k = \frac{p_{lk}}{W(b_l, g_k)}$ and $\hat{f}(b_i, g_k) = \sum_{j \in [m]} \sum_{l \in [n]} f(b_{ij}, g_{lk})$. Then $\hat{p}$ and $\hat{f}$ are the market clearing price vector and corresponding money flow for $AD_G(n, m)$.

(b) If there exists an algorithm that solves $AD_S(n)$ in $T(n)$, then there exists an algorithm that solves $AD_G(n, m)$ in $T(n \cdot m)$. 