1. **Extreme Point Solutions**

   (2) implies (1) is very easy.

   For (3) implies (2), if you were in class, what Joseph said is the solution; if you were not in class, consider the tight constraints.

2. **Minimum s-t Cut**

   My proof is similar to the integrality proof of bipartite vertex cover in class, by showing that a solution is a convex combination of two solutions if it has no integral d_{uv}.

3. **Balanced Coloring**

   Use the iterative approach, and argue that there is some "good" hyperedge if $-1 < x_v < 1$ for all $v \in V$, which should follow from a simple rank argument.

4. **Cooperative Game Theory**

   It is just a straightforward application of LP duality. The conditions are not particularly simple but they are necessary and sufficient.

5. **Multiplicative Weights Update Method**

   Just use the multiplicative weights update method on the capacity constraints, and treat other equality constraints as easy constraints, and then use a shortest path algorithm to implement the oracle.

6. **Sum of Squares and Minimization**

For part (a), think of each $q_i(x)^2$ as a rank one PSD matrix.

For part (b), factorize the polynomial into quadratic polynomials, and recall that complex roots come as conjugate pairs.

Part (c) is easy.

7. **Minimum Multicut**

   The steps are already clearly outlined, with the hint of using Jensen's inequality. You can assume $k \leq |E|$.

8. **Theta Function**

For part (a), write the vector program in matrix form, construct the dual, and go back to the vector form. It is fairly straightforward, but don't forget the zero entries.

For part (b), you should have a vector u_0 with the constraint $\langle u_0, u_o \rangle = t$ in the dual program where t is the objective value of the dual program. The idea is simple: just get rid of this u_0, by subtracting its contribution in other vectors, and then rescale the other vectors to length one. There will be some calculations, which could be simplified if you assume $u_0 = (\sqrt{t}, 0, \ldots, 0)$.