Algorithms - Solution 7

1. In each iteration we find an assignment that satisfies at least half of the equations with the variables $y_1 \dots y_k$. Notice that we work over Z_2, so each equation is either $\sum y_i = 1$ or $\sum y_i = 0$. We choose random assignment- assume it satisfies p equations. Change the assignment of the last variable y_k from 0 to 1 or vice versa. Each equation that was false before is true now and vice versa, so this assignment satisfies $k-p$ equations, and $max(p, k-p) \geq k/2$. Now assume that in each iteration we remove a_i equations from which b_i are satisfied. It is easy to see, that we remove each equation once, that is $\sum a_i = m$. As we just proved, for each i, $b_i \geq a_i/2$. So we have $ALG = \sum b_i \geq \sum a_i/2 = m/2 \geq OPT/2$.
   Now notice that we have $O(m)$ loops in each we scan $O(m)$ equations, each has $O(n)$ variables. So we have that total run time is $O(m^2 n)$. Notice that the bound is tight- look at the following system (for the case w.l.o.g $n+1=m$)
$$\begin{cases} x_1 = c_1 \\ x_1 + x_2 = c_2 \\ x_1 + x_2 + x_3 = c_3 \\ \dots \\ x_1 + x_2 + \dots + x_{n-1} = c_{m-1} \\ x_2 + x_3 + \dots + x_n = c_m \end{cases}$$

2. (a) $f(X,Y) = \sum_{u\in X, v\in Y} f(u,v) = \sum_{u\in X, v\in Y} -f(v,u) = -\sum_{u\in X, v\in Y} f(v,u) = -f(Y,X)$
   (b) $f(X,X) = \sum_{u\in X, v\in X} f(u,v) = \sum_{u\in X,(v\neq u)\in X} \Big(f(u,v) + f(v,u)\Big) + \sum_{u\in X} f(u,u) = 0 + 0 = 0$
   (c) $f(X\cup Y, Z) = \sum_{u\in X\cup Y, v\in Z} f(u,v) \overset{X\cap Y=\emptyset}{=} \sum_{u\in X, v\in Z} f(u,v) + \sum_{u\in Y, v\in Z} f(u,v) = f(X,Z) + f(Y,Z)$
   $f(Z, X\cup Y) = f(Z,X) + f(Z,Y)$ is similar

3. (a) We need to show that the function $g(e) = (1-t)f_1(e) + tf_2(e)$ is a flow, i.e. it fulfills the 3 constraints of a flow: capacity, anti-symmetry and flow preservation.
      i. f_1 and f_2 are flows, i.e. $f_1(e) \leq c(e)$ and $f_2(e) \leq c(e) \Rightarrow$
      $g(e) = (1-t)f_1(e) + tf_2(e) \leq (1-t)c(e) + tc(e) = c(e)$
      ii. $f_1(u,v) = -f_1(v,u)$ and $f_2(u,v) = -f_2(v,u) \Rightarrow$
      $g(u,v) = (1-t)f_1(u,v) + tf_2(u,v) = (1-t)(-f_1(v,u)) + t(-f_2(v,u)) =$
      $-(1-t)(f_1(v,u) - tf_2(v,u) = -\Big((1-t)f_1(v,u) + tf_2(v,u)\Big) = -g(v,u)$
      iii. $\forall (v \neq s,t) \in V : \sum_{u\in V} f_1(u,v) = 0$ *and* $\sum_{u\in V} f_2(u,v) = 0 \Rightarrow$
      $\sum_{u\in V} g(u,v) = \sum_{u\in V} \Big((1-t)f_1(u,v) + tf_2(u,v)\Big) =$
      $\sum_{u\in V}(1-t)f_1(u,v) + \sum_{u\in V} tf_2(u,v) = (1-t)\sum_{u\in V} f_1(u,v) + t\sum_{u\in V} f_2(u,v) =$
      $(1-t)0 + t0 = 0$

(b) LP formulation for MAX-FLOW:
$max\ f(s,V) = \sum_{v \in V} f(s,v)$
$s.t.\ \forall (u,v \in V) : f(u,v) = -f(v,u);\ \ f(u,v) \leq c(u,v)$
$\forall (v \neq s,t) \in V : \sum_{u \in V} f(u,v) = 0$

4. It turns out that we do not have a simple solution for this question. It can be proven using the fact that flows form a convex set. But it is too advanced. Therefore we cancel the question (and will probably give an additive factor on this exercise).

5. We will prove that the value $f(u,v)$ is integer by induction on number of steps of Ford-Fulkerson algorithm. Induction base (step 0) is trivial. Induction hypothesis: assume that after step n the flow on each edge is integer. We find the augmenting path, and the flow on this path is maximal capacity of one of the edges on the path, which, by induction hypothesis, is integer. The Δp on each edge is the difference between the previous flow value and this capacity, and it is integer.
This proves that after each iteration the value on each edge is integer, and, in particular, after the last iteration. From this immediately follows that the max flow value, $\sum_{v \in V} f(s,v)$, is integer.

6. We convert the given graph into flow network, and the existence of the perfect matching is equivalent to existence of the flow of size n (number of vertices on one side of the bipartite graph). According to MAX-FLOW MIN-CUT theorem it is enough to prove that if for each $A \subseteq L, |A| \leq |N(A)|$ then there exists cut of size n. Look at the following cut: the source s, $n-k$ vertices in L, and $n-m$ vertices in R. In this cut there are k edges from s to L and $n-m$ edges from R to t. Let's see how many edges are from L to R. Its it given that the $n-k$ vertices in L have at least $n-k$ neighbors in R. Assume that all $n-m$ vertices in R which are inside the cut are among them, but there are at least $(n-k)-(n-m)$ vertices outside the cut so this is the amount of edges from L to R crossing the cut. The total number of edges in the cut is at least $k+(n-m)+(n-k)-(n-m) = n$.