COMP 360 - Fall 2011 - Sample Final Exam

1. (10 Points) Consider a flow network $(G, s, t, \{c_e\})$.

   (a) Prove or Disprove: A maximum flow f might assign non-integer flows to some edges?

   Solution: The statement is true, and it is easy to construct an example........

   (b) Prove or Disprove: Multiplying all the capacities by 2 will multiply the value of the maximum flow by 2.

   Solution: The statement is true. Let G denote the original flow network and H denote the one obtained after multiplying the capacities by 2. Let c be the value of the maximum flow for G. First note that if f is a flow in G of value c, then $2f$ is a valid flow in H with value $2c$. So $\text{Maxflow}(H) \geq 2\text{Maxflow}(G) = 2c$.

   On the other hand by the mincut-maxflow theorem there is a cut (A, B) in G of capacity c. Now (A, B) has capacity $2c$ in H, and so $\text{Maxflow}(H) \leq 2c$.

   We conclude that $\text{Maxflow}(H) = 2c$.

2. (5 Points) Let $G = (V, E)$ be graph and s, t be two vertices in G. Write the dual of the following linear program:

$$
\begin{array}{lll}
\min & x_t & \\
\text{s.t.} & x_u = x_v & \forall\, uv \in E \\
 & x_s = 1 & \\
 & x_u \geq 0 & \forall\, u \in V
\end{array}
$$

Solution: First we write the LP in the following from:

$$
\begin{array}{lll}
\min & x_t & \\
\text{s.t.} & x_u - x_v = 0 & \forall\, uv \in E \\
 & x_s = 1 & \\
 & x_u \geq 0 & \forall\, u \in V
\end{array}
$$

Now to write the dual, for every $e = uv \in E$ we assign a variable y_e to the constraint $x_u - x_v = 0$, and furthermore we will have one variable z for the constraint $x_s = 1$. Then the dual is

$$
\begin{array}{lll}
\max & z & \\
\text{s.t.} & \sum_{u_0v \in E} y_{u_0v} - \sum_{vu_0 \in E} y_{vu_0} \leq 0 & \forall\, u_0 \in V \setminus \{s, t\} \\
 & \sum_{tv \in E} y_{tv} - \sum_{vt \in E} y_{vt} \leq 1 & \\
 & z + \sum_{sv \in E} y_{sv} - \sum_{vs \in E} y_{vs} \leq 0 & \\
 & \text{all variables are universal} &
\end{array}
$$

3. (5 Points) What is the optimal value to the linear program in the previous question?

   Solution: It is equal to 1 if s and t are in the same component and 0 otherwise.

4. (10 Points) Show that the following problem belongs to P:

$$X = \{\langle G\rangle \mid \text{It is possible to remove exactly 2 vertices from } G \text{ to make it disconnected.}\}.$$

Solution: We check all the $\binom{n}{2}$ possible ways of choosing the 2 vertices, and for each one, we eliminate the two vertices and run a DFS to see whether the graph is disconnected. This is a polytime algorithm as $\binom{n}{2} = O(n^2)$ and that DFS is polytime.

5. (5 Points) Give an example of a PSPACE-complete problem.

   Solution: QSAT. See the textbook.

6. (5 Points) Show that the following language is NP-complete:

$$X = \{\langle \phi\rangle \ : \ \phi \text{ is a CNF which is satisfiable with at least 2 different assignments}\}.$$

Solution: First note that the problem is in NP. Indeed if $\langle\phi\rangle \in X$, then the two different assignments satisfying ϕ are certificates, and their validity can be verified in polytime.

To prove the completeness we reduce SAT to X. Let $\langle\psi\rangle$ be an instance of the SAT problem. Let y be a new variable that does not appear in ψ and set $\phi := \psi \wedge (y \vee \overline{y})$. Note that any satisfying assignment to ψ can be extended to a satisfying assignment to ϕ in two different ways (setting $y = true$ or $y = false$). Thus

$$(\psi \text{ is satisfiable}) \iff (\phi \text{ has at least two different solutions}).$$

So an oracle for X can be used to decide whether ψ is satisfiable efficiently.

7. (10 Points) Show that the following language is NP-complete:

$$X = \{\langle G\rangle \mid G \text{ has an independent set that contains at least half of the vertices}\}.$$

You can use the fact that the following problem is NP-complete:

$$Y = \{\langle G, k\rangle \mid G \text{ has an independent set of size } k \text{ where } k \leq |V(G)|/2\}.$$

Solution: First note that the problem is in NP. Indeed if $\langle G\rangle \in X$, then an independent set of size at least $|V(G)|/2$ is a certificate, and its validity can be verified in polytime.

To prove the completeness we reduce Y to X. Let $\langle G, k\rangle$ be an instance of Y. Let n denote the number of vertices in G. Construct the graph H to be the union of G and m isolated vertices where m is a number to be determined later. We want to satisfy the following property:

$$(\text{Ind}(G) \geq k) \iff \left(\text{Ind}(H) \geq \frac{|V(H)|}{2}\right).$$

Note that H has $n+m$ vertices, and $\text{Ind}(H) = \text{Ind}(G) + m$. So

$$\left(\text{Ind}(H) \geq \frac{|V(H)|}{2}\right) \iff \left(\text{Ind}(G) + m \geq \frac{n+m}{2}\right) \iff \text{Ind}(G) \geq \frac{n-m}{2}.$$

Now taking $m = n - 2k \geq 0$, we have $\frac{n-m}{2} = k$ which shows

$$\left(\text{Ind}(H) \geq \frac{|V(H)|}{2}\right) \iff \text{Ind}(G) \geq k.$$

So $\langle H\rangle \in X$ if and only if $\langle G, k\rangle \in Y$.

8. (10 Points) A path of length k contains $k+1$ vertices. Show that the following language is NP-complete:

$X = \{\langle G, k\rangle \mid$ At least k vertices are needed to be removed from G to eliminate all paths of length $2\}$.

(Hint: Reduce vertex cover to X.)

Solution: First note that the problem is in NP. Indeed if $\langle G, k\rangle \in X$, then the set of k vertices whose removal eliminates all paths of length 2 is a certificate, and its validity can be verified in polytime: We remove those vertices and verify that no vertex has more than one neighbour.

To prove the completeness we reduce VertexCover to X. Let $\langle G, k\rangle$ be an instance of the vertex cover. Construct the graph H from G by attaching a pendant edge to every vertex in G. First note that if S is a vertex cover in G, then removing the vertices of S from H leaves no path of length 2 in H.

On the other hand let T be a minimum set of vertices whose removal eliminates all paths of length 2 in H. Without loss of generality we can assume that all vertices in T are in the original graph G (why?). Then T must be a vertex cover for G as otherwise there will be an edge in G whose endpoints are not removed by T and that leaves a path of length 2 in H (using the pendant edges).

We conclude that the size of the minimum vertex cover in G is equal to the minimum number of vertices required to be removed from H to eliminate all paths of length 2. So

$$\langle G, k\rangle \in \text{VertexCover} \Longleftrightarrow \langle H, k\rangle \in X,$$

and thus an oracle for X can be used to solve the vertex cover problem efficiently.

9. (15 Points) Consider the following optimization version of the SubsetSum problem: Given positive integers $\{w_1, \ldots, w_n\}$ and a positive integer m. We want to find a set $S \subseteq \{w_1, \ldots, w_n\}$ such that $\sum_{w\in S} w \leq m$ and this sum is maximized. Show that the following is a 2-factor approximation algorithm:

- Set $S := \emptyset$.
- Sort the numbers such that $w_1 \geq w_2 \geq \ldots \geq w_n$.
- For $i = 1, \ldots, n$:
-     if it is possible add w_i to S without violating $\sum_{w\in S} w \leq m$, then add w_i to S.

Solution: First note we can discard all $w_i > m$ as they do not affect the optimal solution or the output of the algorithm. Thus we assume that $w_i \leq m$ for $i = 1, \ldots, m$. Let w_t be the first element that is not added to S by the algorithm. Then we know that $w_{t-1} \in S$ and that $\sum_{w\in S} w \geq m - w_t$ as otherwise w_t would be added to S. Also since $w_{t-1} \in S$ we have $\sum_{w\in S} w \geq w_{t-1} \geq w_t$. These two inequalities show that $2\sum_{w\in S} w \geq m$. Thus

$$\sum_{w\in S} w \geq m/2 \geq \text{Opt}/2.$$

10. (10 Points) Let $G = (V, E)$ be a graph. Recall that $S \subseteq V$ is a vertex cover if and only if $V - S$ is an independent set. Also recall that the following is a 2-factor approximation algorithm for vertex cover: Pick any maximal matching M in G and let S be the set of all vertices involved in M. Output S.

Is it true that the following is a $\frac{1}{2}$-factor approximation algorithm for the maximum independent set problem? Pick any maximal matching M in G and let S be the set of all vertices involved in M. Output $V - S$.

Solution: No it is not true. For example if G is a single edge, then the algorithm outputs 0 while the size of the largest independent set is 1 (and $0 \leq \frac{1}{2}$).

11. (10 Points) Let G be a 4-regular graph on n vertices (4-regular means that every vertex is adjacent to 4 edges). We want to color the *edges* of G with two colors Red and Blue such that the number of vertices that are adjacent to exactly two Red and two Blue edges is maximized. If we color the edges at random, then what is the expected number of vertices that satisfy the above condition?

Solution: For a every vertex $v \in G$, define the random variable

$$X_v = \begin{cases} 1 & \text{two red and two blue edges are incident to } v \\ 0 & \text{otherwise} \end{cases}$$

Note that $\sum_{v \in V} X_v$ is the number of vertices that are adjacent to exactly two Red and two Blue edges. Thus the expected number of vertices that satisfy this condition is $\mathbb{E}[\sum_{v \in V} X_v] = \sum_{v \in V} \mathbb{E}[X_v]$. Now

$$\mathbb{E}[X_v] = \Pr[X_v = 1] \times 1 + \Pr[X_v = 0] \times 0 = \Pr[X_v = 1] = \binom{4}{2} \left(\frac{1}{2}\right)^4,$$

where in the last equality we used the fact that there are $\binom{4}{2}$ ways to color the 4 edges incident to v so that exactly two of them are Red and two Blue, and that each such coloring happens with probability $\left(\frac{1}{2}\right)^4$. We conclude that the expected number of vertices that satisfy the desired condition is $n\binom{4}{2}\left(\frac{1}{2}\right)^4 = \frac{3n}{8}$.