COMP 360 - Fall 2019 - Sample Final Exam

1. Either prove that the following problem is NP-complete, or show that it belongs to P:

   - Input: A graph G.
   - Question: Is G a Hamiltonian *bipartite* graph?

   Solution: This problem is NP-complete. A Hamiltonian cycle can be used as a certificate for a YES input and whether it is a Hamiltonian cycle and whether the graph is bipartite can be verified in polynomial time easily.

   To prove the completeness we reduce the Hamiltonian cycle problem to this problem. Consider an input G for the Hamiltonian cycle problem. For every vertex v of G, we add four vertices v^-, v_+, v_-, v^+ together with the path $v^-, v_+v_-v^+$ to the new graph G'. Furthermore if uv is an edge in the original graph G, then we add the edges u^-v^+ and v^+u^- to G'. Note that a Hamiltonian cycle in the original graph can be easily translated to a Hamiltonian cycle in G'. Furthermore consider a Hamiltonian cycle in G'. Since the only way to visit v_+ is through the path $v^-v_+v_-v^+$ (and a Hamiltonian cycle has to visit all the vertices), these three vertices must appear together on the cycle (either as $v^-v_+v_-v^+$ or as $v^+v_-v_+v^-$). As a result any Hamiltonian cycle in G' will correspond to a Hamiltonian cycle in the original graph G. Hence G has a Hamiltonian cycle if and only if G' has a Hamiltonian cycle. Moreover note that G' is bipartite. Hence we can run the oracle on G' to see whether G is Hamiltonian or not.

2. Either prove that the following problem is NP-complete, or show that it belongs to P:

   - Input: A CNF ϕ.
   - Question: Is there a truth assignment that satisfies none of the clauses in ϕ.

   Solution I: This problem is in P. Note that if a variable x_i and its negation $\overline{x_i}$ both appear in the formula then the answer is NO (as in every assignment one of them will be TRUE). Otherwise every variable appears either as x_i or in the negated form $\overline{x_i}$ (but not both). We can decide the value of x_i accordingly ($x_i = F$ if x_i appears, and $x_i = T$ if $\overline{x_i}$ appears) , and this will not satisfy any clauses. Hence the following algorithm solves the problem: Check to see whether there is a variable x_i such that both x_i and $\overline{x_i}$ appear in the formula. In this case output NO, otherwise output YES.

   Solution II: This problem is in P. Note that every clause will uniquely determine the value of all the variables that are involved in that clause (since all the terms in the clause must be false). Hence we can start from the first clause and set the values of the variables accordingly. If at any point we reach a <u>term</u> that has already been set to TRUE, then we terminate and output NO. Otherwise when all the clauses are processed we output YES.

3. Consider the following optimization version of the Subset-Sum problem: Given positive integers $\{w_1, \ldots, w_n\}$ and a positive integer m. We want to find a set $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i \leq m$ and is maximized. Show that the following is a $\frac{1}{2}$-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 i to S without violating $\sum_{i \in S} w_i \leq m$, then add i to S.

Solution: This follows just by definition of algorithm. Suppose $W^* \leq m$ is the optimal solution and the algorithm picked $w_1 \geq w_2 \geq \cdots \geq w_k$ and stopped. We show that $\sum_{i=1}^{k} w_i \geq W^*/2$. Suppose otherwise, then $w_{k+1} \leq w_k < W^*/2$ and hence $\sum_{i=1}^{k+1} w_i < W^* \leq m$, which shows that the algorithm should not have stopped after adding w_k, a contradiction.

4. Problem 10 of Chapter 11 textbook: Suppose you are given an $n \times n$ grid graph G. Associated with each node v is an integer weight $w(v) \geq 0$. You may assume that all the weights are distinct. Your goal is to choose an independent set S of nodes of the grid, so that the sum of the weights of the nodes in S is as large as possible. (The sum of the weights of the nodes in S will be called its total weight.) Consider the following greedy algorithm for this problem.

   - Start with $S := \emptyset$.
   - While some node remains in G:
     - Pick a node v of maximum weight.
     - Add v to S.
     - Delete v and its neighbors from G
   - Endwhile.

   Show that this algorithm returns an independent set of total weight at least $\frac{1}{4}$ times the maximum total weight of any independent set in the grid graph G.

   Solution: Since for every node v picked we remove the neighbors, the algorithm will not output any connected nodes thus the algorithm gives an independent set. Suppose that we pick a node v at some point in the algorithm. Let $v_1, \ldots, v_4$ be its neighbours. Note that none of $v_1, \ldots, v_4$ have been picked at this point (otherwise v would have been deleted). Since v has the maximum weight among the remaining vertices, we have weight$(v) \geq$ weight(v_i) for $i = 1 \ldots 4$. So

   $$4 \times \text{weight}(v) \geq \sum_{i=1}^{4} \text{weight}(v_i).$$

   If the optimal algorithm doesn't choose v and chooses a subset (or all four) of the neighbors instead, then it could be at most 4 times better.

5. Consider a directed bipartite graph $G = (V, E)$. We want to eliminate all the directed cycles of length 4 by removing a smallest possible set of vertices.

   (a) Let $\mathcal{C}_4$ denote the set of all cycles of length 4 in the graph. Show that the following integer program models the problem:

   $$\begin{array}{lll} \min & \sum_{v \in V} x_v & \\ \text{s.t.} & \sum_{u \in C} x_u \geq 1 & \forall C \in \mathcal{C}_4 \\ & x_u \in \{0, 1\} & u \in V \end{array}$$

   Solution: For each vertex v, we have a variable x_v. These variables are 0/1 valued. The meaning of $x_v = 1$ is that we remove vertex v from the graph. The meaning of $x_v = 0$ is that we keep vertex v. Let OPT denote the optimum value for the original problem. Let OPT_{ip} denote the optimum value for the integer program. Let x^* be an optimum solution of the integer program. By the inequality constraint, the integer program will pick at least one vertex from each 4-cycle. Thus removing the vertices corresponding to $x^* = 1$ will remove all the 4-cycles. Therefore we have $\mathrm{OPT} \leq \mathrm{OPT}_{ip}$. On the other hand, take a minimum set of vertices whose removal kills all the 4-cycles. Setting $x_v = 1$ for these vertices clearly produces a feasible solution for the integer program. Therefore $\mathrm{OPT}_{ip} \leq \mathrm{OPT}$.

(b) Why does the optimal solution to the following relaxation provides a lower bound for the optimal answer to the above integer linear program? In other words why it is not necessary to have the constraints $x_u \leq 1$ in the relaxation?

$$\begin{array}{lll} \min & \sum_{v\in V} x_v & \\ \text{s.t.} & \sum_{u\in C} x_u \geq 1 & \forall C \in \mathcal{C}_4 \\ & x_u \geq 0 & \forall u \in V \end{array}$$

Solution: We claim that in any optimum solution x^*, $x^*_u \leq 1$ for all u. Suppose there exists some u such that $x^*_u > 1$. Round down the value of this variable to 1. Note that all the inequality constraints will still be satisfied. So we still have a feasible solution. On the other hand, the optimum value will go down, which is a contradiction.

(c) Give a simple 4-factor approximation algorithm for the problem based on rounding the solution to the above linear program.

Solution: As before, let x^* be the optimum solution. The rounding is as follows. If $x^*_u \geq 1/4$, set $x^*_u = 1$, otherwise set $x^*_u = 0$. First let's check that we get a feasible solution to our problem. In each inequality constraint, it must be the case that at least one of the variables has value $\geq 1/4$. Thus in our rounded solution, we pick at least one vertex from each 4-cycle. So we kill all the 4-cycles as required. Let OPT^* be the optimum for the linear program, let OPT be the optimum for the original problem and let A be the value obtained by rounding the optimum of the linear program. Clearly $\mathrm{OPT}^* \leq \mathrm{OPT}$. Also, by our rounding scheme, we have $A \leq 4\mathrm{OPT}^*$. Thus, $A \leq 4\mathrm{OPT}$, i.e. our solution is within a factor 4 of the optimum.

(d) (Let L and R denote the set of the vertices in the two parts of the bipartite graph. (Every edge has one endpoint in L and one endpoint in R). Let x^* denote an optimal solution to the linear program in Part (b). We round x^* in the following way:

For every $u \in V$,

- if $u \in R$ and $x^*_u \geq 1/2$, set $\widehat{x}_u = 1$.
- if $u \in L$ and $x^*_u > 0$, set $\widehat{x}_u = 1$.
- Otherwise set $\widehat{x}_u = 0$.

Show that $\widehat{x}$ is a feasible solution to the integer linear program.

Solution: Observe that each 4-cycle contains two vertices from L and two vertices from R. Consider an inequality constraint of the linear program (so we are considering a fixed 4-cycle). If $x^*_u > 0$ for one of the two vertices in L, $\widehat{x}_u$ will be set to 1 and therefore this inequality will be satisfied. On the other hand, if $x^*_u = 0$ for both vertices in L, then it must be the case that $x^*_v \geq 1/2$ for one of the vertices in R. Thus this vertex will be rounded to 1 and the inequality will be satisfied.

(e) Consider the dual of the relaxation:

$$\begin{array}{lll} \max & \sum_{C\in\mathcal{C}_4} y_C & \\ \text{s.t.} & \sum_{C\in\mathcal{C}_4, u\in C} y_C \le 1 & \forall u \in V \\ & y_C \ge 0 & \forall C \in \mathcal{C}_4 \end{array}$$

and let y^* be an optimal solution to the dual. Use the complementary slackness to prove the following statement: For every $C \in \mathcal{C}_4$ either we have $|\{u : \widehat{x}_u = 1\}| \le 3$ or $y^*_C = 0$.

Solution: Suppose $|\{u : \widehat{x}_u = 1\}| > 3$. Then all the variables for that cycle must be rounded to 1. For that to happen, it must be that $x^*_u \ge 1/2$ for the vertices in R and $x^*_u > 0$ for the vertices in L. Thus, we must have $\sum_{u\in C} x^*_u > 1$, i.e. the constraint is not tight. By complementary slackness, this means $y^*_C = 0$.

(f) Use the complementary slackness and the previous parts to show that our rounding algorithm is a 3-factor approximation algorithm.

Solution: As mentioned before, we have $\sum_{u\in V} x^*_u = \text{OPT}^* \le \text{OPT}$. Thus, we are done once we show

$$\sum_{u\in V} \widehat{x}_u \le 3\text{OPT}^*.$$

Note that if $\widehat{x}_u = 1$, $x^*_u > 0$. Therefore, by complementary slackness, $\sum_{C\in\mathcal{C}_4, u\in C} y^*_C = 1$. The variables $\widehat{x}_u$ are 0/1 valued, so we can write

$$\sum_{u\in V} \widehat{x}_u = \sum_{u\in V} \widehat{x}_u \sum_{C\in\mathcal{C}_4, u\in C} y^*_C = \sum_{u\in V} \sum_{C\in\mathcal{C}_4, u\in C} \widehat{x}_u y^*_C.$$

We now change the order of the sums and get

$$\sum_{u\in V} \sum_{C\in\mathcal{C}_4, u\in C} \widehat{x}_u y^*_C = \sum_{C\in\mathcal{C}_4} \sum_{u\in C} \widehat{x}_u y^*_C = \sum_{C\in\mathcal{C}_4} y^*_C \sum_{u\in C} \widehat{x}_u.$$

From part (e) of the question, we know that if $y^*_C \ne 0$, then $\sum_{u\in C} \widehat{x}_u \le 3$. Therefore the above quantity can be upper bounded by $3\sum_{C\in\mathcal{C}_4} y^*_C = 3\text{OPT}^*$ (the equality follows from duality). Putting things together, we have shown

$$\sum_{u\in V} \widehat{x}_u \le 3\text{OPT}^*$$

as required.