HW 8 CMSC 452. Morally Due April 23
THIS HW IS TWO PAGES LONG!!!!!!!!!
SOLUTIONS

1. (30 points) A *poly inequality* is an inequality of the form

$$p(x_1, x_2, \ldots, x_n) \le c$$

where $p(x_1, \ldots, x_n)$ is a polynomial with integer coefficients WITHOUT a constant term, and $c \in \mathsf{Z}$.

TWO EXAMPLES:

$$x_1^3 x_4^2 - 2x_2x_3 + 18x_3^{14}x_4^2 + x_1 \le 1000.$$

$$x_1 + x_2 \le 89$$

Let POLY PROGRAMMING, called PP, be the following problem:

Given a set of poly inequalities determine if there is some way to set the variables to rationals so that all the inequalities hold.

(a) Show that 3-$SAT \le PP$.

(b) Use your reduction on the following formula (i.e., list the inequalities produced by the reduction)

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4)$$

GO TO THE NEXT PAGE!!!!!!!!!!!!

SOLUTION TO PROBLEM ONE

(a) Given a formula $\phi(x_1, \ldots, x_n)$ output the following constraints

- For all i, $x_i^2 - x_i \le 0$ and $x_i^2 - x_i \ge 0$. Hence we have $x_i^2 = x_i$ so $x_i = 0$ or $x_i = 1$.
- We now just use the same constraints based on clauses that we did for IP. For each clause $(L_1 \vee L_2 \vee L_3)$ do the following: If L_i is a variable (not a negation of one) then let M_i be that variable (e.g., x_{18}). If L_i is a negation of a variable then let M_i be $1-$ that variable, (e.g., $1 - x_{18}$).
  Add the following linear inequality to the constraints:

$$M_1 + M_2 + M_3 \ge 1$$

(b) We first write the equations as they occur naturally, and then we put them in the right form.

$x_1^2 - x_1 \le 0$
$x_1^2 - x_1 \ge 0$: REWRITE $-x_1^2 + x_1 \le 0$
$x_2^2 - x_2 \le 0$
$x_2^2 - x_2 \ge 0$: REWRITE $-x_2^2 + x_2 \le 0$
$x_3^2 - x_3 \le 0$
$x_3^2 - x_3 \ge 0$: REWRITE $-x_3^2 + x_3 \le 0$
$x_4^2 - x_4 \le 0$
$x_4^2 - x_4 \ge 0$: REWRITE $-x_4^2 + x_4 \le 0$
$x_1 + (1 - x_2) + x_3 \ge 1$: REWRITE $-x_1 + x_2 - x_3 \le 0$
$(1 - x_1) + x_2 + x_4 \ge 1$: REWRITE $x_1 - x_2 - x_4 \le 0$
$x_2 + (1 - x_3) + (1 - x_4) \ge 1$: REWRITE $-x_2 + x_3 + x_4 \le 1$

2. (40 points) Let

$$\text{CLIQ17} = \{G \mid \text{ graph } G \text{ has a clique of size 17 }\}$$

(a) (25 points) Either show that CLIQ17 is in P or show that CLIQ17 is NP-complete or do both. (ALSO — not to hand in, but think about — is it likely that someone in the class will be able to do both?)

(b) (25 points) Is CLIQ17 closed under minors (see Wikipedia entry for clarification). That is, if $G \in$ CLIQ17 and H is a minor of G, is it necessarily true that $H \in$ CLIQ17? If so then prove it, if not then give a counterexample.

`https://en.wikipedia.org/wiki/Graph_minor`

SOLUTION TO PROBLEM TWO

a) CLIQ17 is in P:

- Input G
- For ALL sets of vertices of size 17 check if they form a clique. If any do then output YES, else NO.

This is in poly time since the number of sets of cliques to check is $\binom{n}{17} \le n^{17}$.

THINK ABOUT PART: if someone proved that CLIQ17 was both in P and NP-complete then this would imply $P = NP$. This is unlikely to be true and unlikely to be proven by *anyone* at this time. But HEY — you never know!

b) CLIQ17 is NOT closed under minors. Take K_{17}. It is IN CLIQ17. Remove one vertex. Now it is K_{16} which is NOT in CLIQ17.

3. (30 points) Let

$$FACT = \{(n, x) \mid \text{ there is a nontrivial factor of } n \text{ that is } \le x \ \}.$$

(A NONTRIVIAL factor of n is a positive factor that is NOT 1 and NOT n.)

n and x are both positive integers and are given in binary, so the NUMBER (say, for example) ONE THOUSAND only takes around 10 bits, NOT 1000 bits, to input.

Let $FFACT$ be the function that, on input n, outputs the complete prime factorization of n.

Show that if $FACT \in P$ then $FFACT$ can be computed in Polynomial time.

NOTE- poly in the LENGTH of the input. So the LENGTH of ONE THOUSAND would be TEN. So $FACT \in P$ means that it takes time $p(\log n + \log x)$ to decide (n, x) for some poly p.

SOLUTION TO PROBLEM THREE

Note that if n has a nontrivial factor then it has one $\le \lceil \sqrt{n} \rceil$. We use this. If you used n instead, that would be fine also.

Assume $FACT \in P$.

ALGORITHM FOR FFACT. We will be calling it recursively.

(a) Input n (assume $n > 1$)

(b) Ask $(n, \lceil \sqrt{n} \rceil) \in FACT$? If NO then n is prime so output n.

(c) If we got here then $(n, \lceil \sqrt{n} \rceil) \in FACT$. So we know there is a nontrivial factor of n between 2 and $\lceil \sqrt{n} \rceil$. Do a binary search using queries to FFACT to find the LEAST such factor m_1. The number of queries is $O(\log \sqrt{n}) = O(\log n)$. Each one takes $O(p(\log n))$ steps. So $O(\log(n) \cdot p(\log n))$ time is taken.

(d) Note that m_1 is prime, and $m_2 = n/m_1$ is an integer.

(e) Call FFACT on m_2. Output m_1 in addition to the factorization of m_2 produced.

There will be only $O(\log n)$ recursive calls to FFACT. Hence, the total running time is $O(\log(n)^2 \cdot p(\log n))$, which is polynomial in terms of $\log n$.