Practice problems

1. Show that that the language $\mathsf{BIPARTITE} = \{G | G \text{ is bipartite } \}$ is in P.
2. Show that the language $\mathsf{Triangle - Free} = \{G | G \text{ is triangle-free } \}$ is in P.
3. Assuming $NP \neq coNP$, prove that no NP-complete problem can be in coNP.
4. Show that the Halting problem is NP-hard.
5. $DOUBLESAT = \{\phi | \phi \text{ is a CNF formula with at least 2 satisfying assignments}\}$. Show that the problem is NP-complete.

Solutions in the next page. Please try the problems first.

1. If a graph is bipartite then it is possible to colour its vertices with 2 colours such that for any edge its endpoints are of different colours: give all vertices of first partition colour red and all vertices of the second partition colour blue; any edge has one endpoint in the red partition and the other endpoint in the blue partition. Now, from a vertex v, build a BFS tree such that v is given the colour red, its neighbours are coloured blue, the vertices in the third level are coloured red and so on - vertices in alternate levels get red and blue. Consider a cross edge in the BFS tree - the endpoints can be in the same level, or at a difference of 1 level. The endpoints will get the same colour if and only if they are in the same level. Therefore, if there are cross edges that have endpoints in the same level then the graph is not bipartite, otherwise it is. Running time $= O(n^2)$ where n is the number of vertices.

2. Enumerate all vertex subsets of size 3. This takes $O(n^3)$ time if n is the number of vertices in the graph. Check for each 3-vertex subset if the vertices form a triangle in the graph. If you find a triangle then the graph is not triangle-free, otherwise it is.

3. If an NP-complete Π problem is in coNP, then there is a polynomial p and a polynomial-time deterministic TM such that on an input string x, for all $z \in \{0,1\}^{p(|x|}$ $M(x,z) = 1$. Now, consider a language $L \in NP$. There is a polynomial time reduction $L \leq_p \Pi$ for a polynomial p', conducted by a TM M_1. We design a TM M_L for L that shows that L is in coNP: M_L first runs M_1 to reduce the input instance x of L to an instance x' of Π such that $|x'| \leq q(|x|)$. Note that $p(|x'|) = O(p(q(|x|)))$, where $r = p \ldots q$ is also a polynomial function. Then M_L runs M on x' and every $z \in \{0,1\}^{r(|x|)}$. Thus, M_L accepts the language L. This implies $NP \subseteq coNP$. On the other hand, $\overline{\Pi}$, which is a coNP-complete problem would be in NP. This means that $coNP \subseteq NP$. Thus, this would imply $NP = coNP$, which is a contradiction.

4. HP is NP–hard if $L \leq_p HP$ for all $L \in NP$. Equivalently HP is NP–hard if $SAT \leq_p HP$. We show a polynomial time computable function mapping a formula ϕ to a TM N and a string x such that $\phi \in SAT$ iff $(N, x) \in HP$. Let M be a polynomial time deterministic machine that takes as input a formula ϕ, an assignment z and accepts iff z satisfies ϕ. Such a TM exists since $SAT \in NP$. Construct N so that on input x it does the following:
(i) Parse x as a Boolean formula over n variables. If x cannot be parsed as a Boolean formula then N enters a trivial loop.
(ii) For all assignments $z \in \{0,1\}^n$, run $M(x,z)$ to check if z satisfies x; accept and halt if M accepts.
(iii) If x is not satisfied by any z, enter a trivial loop.
Map ϕ to (N, ϕ). This map can be computed in time polynomial in $|\phi|$ (the size of ϕ) irrespective of fact that N 's running time would be exponential in the $|\phi|$. Now, $\phi \in SAT$ iff $\exists z$ such that $M(\phi, z)$ accepts iff N halts on input ϕ iff $(N, \phi) \in HP$. This completes the reduction $SAT \leq_p HP$.

5. Reduction from SAT. Take a formula ψ of SAT, and create a formula ϕ for DOUBLESAT by introducing a new variable z and adding an extra clause $(z \vee \neg z)$ to ψ. Notice that ψ has a satisfying assignment if and only if ϕ has two satisfying assignments (one assignment where $z = 1$ and another where $\neg z = 1$).