Problemes Tractables vs Problemes Intractables (Part I)

1. **2-color vs 3-color**. Recordeu que un graf no dirigit $G = (V, E)$ és *bipartit* si existeix una partició de V en V_1, V_2 de manera que tota aresta $uv \in E$ satisfà o bé $u \in V_1 \wedge u \in V_2$ o bé $u \in V_2 \wedge u \in V_1$. Fixeu-vos que dir que un graf és bipartit és equivalent a dir que és 2-colorable (2-COLOR). Definim BIPARTITE GRAPH $= \{\langle G\rangle \mid G$ és bipartit$\}$.

   Demostreu que BIPARTITE GRAPH $\in$ P.

2. **dnf-sat vs cnf-sat**. Demostreu que el problemes següents pertanyen a la classe P:

   a. Donada una fórmula booleana en *Forma Normal Conjunctiva* on cada clàusula té com a màxim dos literals, decidir si és satisfactible.

   b. Donada una fórmula booleana en *Forma Normal Disjunctiva*, decidir si és satisfactible.

3. **Eulerian graph vs Hamiltonian graph** Recordeu que un *recorregut eulerià* en un graf no dirigit és un cicle que pot visitar repetides vegades un mateix vèrtex i ha d'utilitzar cada aresta exactament una vegada. Diem que un graf és *eulerià* si conté un recorregut eulerià.

   Demostreu que

   EULERIAN GRAPH $= \{\langle G\rangle \mid G$ és eulerià$\} \in$ P.

   Dissenyeu un algorisme eficient que en el cas que el graf sigui eulerià ens retorni un recorregut eulerià.

4. **Shortest path vs longest path** Considerem les dues versions següents de *Shortest Path* i *Longest Path*, respectivament, per a grafs no dirigits:

   SPATH $= \{\langle G, a, b, k\rangle \mid G$ conté un camí simple de a a b de longitud menor o igual que $k\}$

   LPATH $= \{\langle G, a, b, k\rangle \mid G$ conté un camí simple de a a b de longitud més gran o igual que $k\}$.

   a. Demostreu que SPATH $\in$ P.

   b. Demostreu que LPATH $\in$ NP. Demostreu que si LPATH $\in$ P aleshores el problema del camí Hamiltonià en graf no dirigits també seria a P.

5. **Half clique.** Considereu el problema SUBGRAF COMPLET D'ORDRE MEITAT (HALF-CLIQUE): Donat un graf no dirigit G, decidir si conté una $\lfloor \frac{n}{2} \rfloor$-clique.

   a. Demostreu que HALF-CLIQUE $\in$ NP

   b. Demostreu que si HALF-CLIQUE $\in$ P, aleshores CLIQUE $\in$ P.

6. **Self reducibility SAT**. Suposem que tenim un algorisme que decideix SAT en temps polinòmic. Dissenyeu un algorisme tal que donada una fòrmula boolena F retorni una assignació que la satisfaci, si existeix. Demostreu que si SAT $\in$ P aleshores *trobar* una assignació que la satisfà també és computable en temps polinòmic. Aquesta propietat s'anomena *self-reducibility*.

7. **Self reducibility CLIQUE**. Suposem que tenim un algorisme que decideix CLIQUE en temps polinòmic. Dissenyeu un algorisme tal que donat un graf G i donat un natural k, retorni una clica de k vèrtexs, si existeix. Demostreu que si CLIQUE $\in$ P aleshores *trobar* una clique també és computable en temps polinòmic. Aquesta propietat s'anomena *self-reducibility*.

8. **Stingy SAT.** Let us consider the following problem:

   STINGY SAT Given a set of clauses (each a disjunction of literals) and an integer k, find a satisfying assignment in which at most k variables are true, if such an assignment exists.

Prove that STINGY SAT is NP-complete.

9. **At most 3.** Consider the CLIQUE problem restricted to graphs in which every vertex has degree at most 3. Call this problem CLIQUE-3.

   (a) Prove that CLIQUE-3 is in NP.

   (b) What is wrong with the following proof of NP-completeness for CLIQUE-3? We know that the CLIQUE problem in general graphs is NP-complete, so it is enough to present a reduction from CLIQUE-3 to CLIQUE. Given a graph G with vertices of degree $\leqslant 3$, and a parameter k, the reduction leaves the graph and parameter unchanged: clearly the output for the reduction is a possible input for the CLIQUE problem. Furthermore, the answer to both problems is identical. This proves the correctness of the reductions, and, therefore, the NP-completeness of CLIQUE-3.

   (c) It is true the the VERTEX COVER problem remains NP-complete even when restricted to graphs in which every vertex has degree at most 3. Call this problem VC-3. What is wrong in the following proof of NP-completeness for CLIQUE-3?

   We present a reduction from VC-3 to CLIQUE-3. Given a graph $G = (V, E)$ with node degrees bounded by 3, and a parameter g, we create an instance of CLIQUE-3 by leaving the graph unchanged and switching the parameter to $|V| - b$. Now, a subset $C \subseteq V$ is a vertex cover in G if and only if the complementary set $V - C$ is a clique in G. Therefore G has a vertex cover of size $\leqslant b$ if and only if it has a clique of size $\geqslant |V| - b$. This proves the correctness of the reduction and, consequently, the NP-completeness of CLIQUE-3.

   (d) Describe an $O(|V|)$ algorithm for CLIQUE-3.

10. **Experimental cuisine.** We are feeling experimental and want to create a new dish. There are various ingredients we can choose from and we'd like to use as many of them as possible, but some ingredients don't go well with others. If there are n possible ingredients (numbered 1 to n), we write down and $n \times n$ matrix D giving the *discord* between any pair of ingredients. This *discord* is a real number between 0.0 and 1.0, where 0.0 means "they go together perfectly" and 1.0 means "they don't go together". For example, if $D[2, 3] = 0.1$ and $D[1, 5] = 1.0$, then ingredients 2 and 3 go together pretty well whereas 1 and 5 clash badly. Notice that D is necessarely symmetric and that the diagonal entries are always 0.0. Any set of ingredients always incurs a *penalty* which is *the sum of all discord values between pairs of ingredients*. For instance the set of ingredients $\{1, 2, 3\}$ incurs a penalty of $D[1, 2] + D[1, 3] + D[2, 3]$. We want this penalty to be small.

    EXPERIMENTAL CUISINE Given n ingredients, the discord $n \times n$ matrix D and some number p, compute the maximum number of ingredients we can choose with penalty $\leqslant p$

    Show that if EXPERIMENTAL CUISINE is solvable in polynomial time, then so is 3SAT.