1 Problemes Tractables vs Problemes Intractables

Remind that

- FPSPACE denotes the class of functions Polynomial Space computable,
- FP denotes the class of functions Polynomial Time computable, and
- co-NP $= \{A | \overline{A} \in \mathsf{NP}\}$

---

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 (CNF)* on cada clàusula té com a màxim dos literals, decidir si és satisfactible.

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

   (c) Una una fórmula booleana en CNF es pot transformar en una fórmula booleana en DNF equivalent? Si es pot, en quant temps?

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

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

   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:

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

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

   (a) Demostreu que $\mathsf{SPATH} \in \mathsf{P}$.

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

5. Consider the problem Modular exponentiation: given $a, b, n \in \mathbb{N}$, compute $d = a^b \bmod n$. Can Modular exponentiation be solved in polynomial time?

6. **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.

7. **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*.

8. **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*.

9. **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.

10. **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 ≤ 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 b, 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 $\leq b$ if and only if it has a clique of size $\geq |V| - b$. This proves the correctness of the reduction and, consequently, the NP-completeness of CLIQUE-3.

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

11. **At most twice.** Show that 3SAT remains NP-complete even when restricted to formulas in which each literal appears at most twice.

12. **Degree at most 4** In the previous exercise we saw that 3SAT remains NP-complete even when restricted to formulas in which each literal appears at most twice.

    (a) Show that if each literal appears at most once, then the problem is solvable in polynomial time.

    (b) Show that INDEPENDENT SET remains NP-complete even in the special case when all nodes in the graph have degree at most 4.

13. Consider a set of linear inequalities where all the coefficients are integers. Consider the problem IntLinEq of deciding if a given set of linear inequalities has an integer solution. Related to this problem we can consider a variant of linear programming Integer Programming in which the veriables are restricted to be integers.

    Show that IntLinEq is NPcomplete and that Integer Programming is NP-hard.

14. Lucas' theorem says the following: If we have an integer a such that: $a^{n-1} \equiv 1 \bmod n$, and, for every prime factor q of $n - 1$, it is not the case that $a^{(n-1)/q} \equiv 1 \bmod n$, then n is prime.

    Can this result be used to show that Primality belongs to NP?

15. Consider the following problems:

- MODULAR FACTORIAL: Given N bits natural numbers x, y compute $x! \mod y$.
- SMALLEST PRIME DIVISOR: Given a N bit natural number x, compute the smallest prime divisor of x.
- FACTORING: Given a N bit natural number x, compute the factorization of x as product of primes.

(a) Prove that y is prime if and only if, for each integer $x < y$, we have that $\mathrm{mcd}(x!, y) = 1$.

(b) Show that if MODULAR FACTORIAL can be solved in polynomial time, then SMALLEST PRIME DIVISOR and FACTORING could be solved in polynomial time.

16. **Proving NP-completeness by generalization.** For each of the problems below, prove that is NP-complete by showing that is a generalization of some classical NP-complete problems.

    (a) SUBGRAPH ISOMORPHISM: Given two undirected graphs G and H, is G a subgraph of H?

    (b) MAX SAT: Given a boolean formula F and an integer k, decide whether there is an assignment that satisfies at least k clauses.

    (c) DENSE SUBGRAPH: Given a graph G and two integers a, b, decide whether there exist a vertices of G such that there are at least b edges between them.

    (d) SPARSE SUBGRAPH: Given a graph G and two integers a, b, decide whether there exist a vertices of G such that there are at most b edges between them.

    (e) RELIABLE NETWORK: We are given to $n \times n$ matrices, a *distance* matrix d_{ij} and a *connectivity requirement* matrix r_{ij}, as well as a budget b; we must find a graph $G(\{1, \ldots, n\}, E)$ such that (1) the total cost of all edges is b or less and (2) between any two distinct vertices i and j there are r_{ij} vertex-disjoint paths.

17. **Feedback set.** Many algorithmic problems are easier to solve or are only well-defined for directed acyclic graphs. For instance, in task scheduling, the edges may represent precedence scheduling constraints, and we need to identify the smallest number of constraints that must be dropped so as to permit a valid schedule; namely, the graph of tasks and constraints does not contain directed cycles. Given a directed graph $G = (V, E)$, a subset E' of E is a *feedback arc set* if by deleting all the edges in E' from G, we obtain an acyclic graph.

    FAS: Given a directed graph $G = (V, E)$ and an integer k, does there exist a feedback arc set of size k?

    Show that FAS is NP-complete.

    (*Hint*: You might consider a reduction from VERTEX COVER. On input (G, k) where G is an undirected graph $G = (V, E)$ and we want to know if it contains a vertex cover of size k. We construct a directed graph $G' = (V', E')$ as follow. Let the vertices of G be $v_1, \ldots v_n$. Then, G' has $2n$ vertices $w_1, w'_1, \ldots, w_n, w'_n$ and edges (w_i, w'_i) for $i = 1, \ldots, n$ and $(w'_i, w_j), (w'_j, w_i)$ for every edge $(v_i, v_j) \in E$.)

18. **Maximum clique.** Let us denote by MAX-CLIQUE the problem defined as follows:

    Given a graph $G = (V, E)$ compute a *clique of maximum size*. That is, compute $U \subseteq V$ such that $\forall u, v \in U$, $u \neq v$, then $(u, v) \in E$ and $\forall U'$, $U' \subseteq V$ with $|U'| > |U|$ then $\exists u', v' \in U'$ such that $(u', v') \notin E$.

    (a) Show that MAX-CLIQUE$\in$ FPSPACE.

    (b) Show that $\mathsf{P} = \mathsf{NP}$ if and only if MAX-CLIQUE$\in$ FP.

19. **Max number of SAT clauses.** Let us denote by MAX-SAT the problem defined as follows:

    Given a boolean formula Φ in Conjunctive Normal Form, compute a boolean assignment a that satisfies the maximum number of clauses. That is, if a satisfies k clauses of Φ, then there is no assignment a' satisfying more than k clauses.

    (a) Show that MAX-SAT $\in \mathsf{FPSPACE}$.

    (b) Show that $\mathsf{P} = \mathsf{NP}$ if and only if MAX-SAT$\in \mathsf{FP}$.

20. **Balanced Colors.** A k-coloring is balanced if exactly $1/k$ of the vertices have each color. Show that for any $k \geq 3$ the problem of whether a graph has a BALANCED k-COLORING is NP-complete. Then show that it is in P for $k = 2$.

21. **Succinct formula.** Remind that we say two boolean formulas are *equivalent* if they have the same set of variables and are true on the same set of assignments to those variables (i.e., they describe the same boolean function). A boolean formula is *minimal* if no shorter Boolean formula is equivalent to it. Let MIN-FORMULA be the collection of minimal Boolean formulas. Show that if $\mathsf{P} = \mathsf{NP}$, then MIN-FORMULA $\in \mathsf{P}$.

22. **Geography game.** One way to pass the time during long journey is to play the word game of *Geography.* The first player names a place, say Barcelona. The second player replies with a place whose name starts with the same letter the previous place ended with, such as Athens. The first player could then respond with Sicily, and so on. Each place can only be named once, and the first player who cannot make a move loses.

We can make a mathematical version of this game. We have a directed graph $G = (V, E)$, and a designated start node $s \in V$. The nodes represent the places and the directed edges represent the legal moves. We start at s, and you and I take turns deciding which step to do next. Whoever ends up in a dead end, where every outgoing edge points to a vertex we have already visited, loses. A position or node v is a *winning position* for the first player if whatever legal move the second player selects, the first player never ends up in a dead end.

GEOGRAPHY

Input: An directed graph $G = (V, E)$ and a start node s.

Question: Is s a *forced* win position for the first player? (The first player with strategy s always wins whatever is the strategy selected by the second player).

Show that GEOGRAPHY is in PSPACE. (In fact GEOGRAPHY is a well known PSPACE-complete problem).

23. **Geography with repeat visits.** Suppose we play GEOGRAPHY without the restriction that each vertex we can only be visited at most once. However, the graph is still directed, and you lose if you find yourself at a vertex with no outgoing edges. Show that the problem of telling whether the first player has a winning strategy in this case is in P. (Notice that if a node is reached after a number of turns greater than $n-1$, then, for sure there is a draw).

24. **Chess.** Consider a board game such as Go or Chess, where the state space of possible positions on an $n \times n$ board is exponentially large and there is no guarantee that the game only lasts for a polynomial number of moves. Assume for the moment that there is no restriction on visiting the same position twice.

   (i) Show that, given configuration of a $n \times n$ board, telling whether the first player of has a *forced* win strategy is in EXP (CHESS or GO are well known EXP-complete problems).

   (iii) Show that if there is a guarantee that the game only lasts a polynomial number of moves, then such a problem will be in PSPACE.