Algorithms – II

Third Short Test

---

1. Let Q_1, Q_2 be two decision problems that admit a polynomial-time reduction $Q_1 \leqslant Q_2$. If P $\neq$ NP, which of the following statements is true?

- **(A)** If Q_1 is NP-Complete, then Q_2 is NP-Complete.
- **(B)** If Q_2 is NP-Complete, then Q_1 is NP-Complete.
- **(C)** If $Q_1 \in$ P, then Q_2 cannot be NP-Complete.
- **[(D)]** If $Q_2 \in$ P, then Q_1 cannot be NP-Complete.

2. Let Q_1, Q_2 be computational problems in NP such that the same (polynomial-time) reduction map f can be used to establish both $Q_1 \leqslant Q_2$ and $Q_2 \leqslant Q_1$. For example, the map $(G = (V,E), k) \mapsto (\overline{G} = (V, \overline{E}), k)$ works for both the reductions CLIQUE $\leqslant$ INDEPENDENT-SET and INDEPENDENT-SET $\leqslant$ CLIQUE. Which of the following statements is true for such a map f in general?

- **(A)** f is a one-to-one correspondence.
- **[(B)]** f may or may not be a one-to-one correspondence.
- **(C)** f is not a one-to-one correspondence.
- **(D)** The existence of f implies that both Q_1 and Q_2 are NP-Complete.

Solution Let EMC (resp. OMC) denote the problem of deciding whether the maximum clique of an undirected graph consists of an even (resp. odd) number of vertices. Given a graph G, add a new vertex, and connect the new vertex to all the existing vertices. This construction works for both EMC $\leqslant$ OMC and OMC $\leqslant$ EMC, but is not bijective, because $f(f(G))$ contains two more vertices than G, and so $f \circ f$ cannot be the identity map.

In order to see that (D) is false, consider the two problems in P.

IS-MAX (resp. IS-MIN): Given an array A of integers and an integer k, decide whether k is the maximum (resp. minimum) element of A.

Look at the reduction $(A, k) \mapsto (-A, -k)$.

3. Which of the following statements is necessarily true? Here n is the input size.

- **(A)** If SAT has an $O(n^t)$-time algorithm (t constant), then every problem in NP has an $O(n^t)$-time algorithm.
- **(B)** If SAT has an $O(2^n)$-time algorithm, then every problem in NP has an $O(2^n)$-time algorithm.
- **(C)** If SAT has an $O(n^{\log n})$-time algorithm, then every problem in NP has an $O(n^{\log n})$-time algorithm.
- **[(D)]** If SAT has an $n^{O(\log n)}$-time algorithm, then every problem in NP has an $n^{O(\log n)}$-time algorithm.

Solution SAT in NP-Complete. Take any $Q \in$ NP. There is a reduction $Q \leqslant$ SAT. If I is an instance of size n for Q, the reduction generates an instance for SAT of size $O(n^k)$ for some positive constant k. If the SAT-solver runs in $f(n)$ time, it runs in $f(n^k)$ time on the converted instance. We may have $k > 1$, so only (D) is necessarily true. For example, $O((n^k)^t) = O(n^{tk})$ is not $O(n^t)$ for $k > 1$, whereas $(n^k)^{O(\log n^k)} = n^{k^2 O(\log n)} = n^{O(\log n)}$, since k is a constant.

4. Let Q_1, Q_2 be two NP-Complete problems. Assume that the input spaces for both the problems are the same. Define the following problems.

$Q_1 \vee Q_2$: Given I, decide whether $I \in \text{Accept}(Q_1)$ or $I \in \text{Accept}(Q_2)$.

$Q_1 \wedge Q_2$: Given I, decide whether $I \in \text{Accept}(Q_1)$ and $I \in \text{Accept}(Q_2)$.

Which of the following statements is true? Assume P $\neq$ NP.

- **[(A)]** Neither $Q_1 \vee Q_2$ nor $Q_1 \wedge Q_2$ may be NP-Complete.
- **(B)** $Q_1 \vee Q_2$ must be NP-Complete, but $Q_1 \wedge Q_2$ may or may not be NP-Complete.
- **(C)** $Q_1 \wedge Q_2$ must be NP-Complete, but $Q_1 \vee Q_2$ may or may not be NP-Complete.
- **(D)** Both $Q_1 \vee Q_2$ and $Q_1 \wedge Q_2$ must be NP-Complete.

Solution Let G be a weighted undirected graph with both positive and negative weights allowed (you may restrict the weights to integer values). Then LONGEST-PATH and SHORTEST-PATH are essentially the same problem, and both are NP-Complete. Define Q_1 as the problem to decide whether a graph G has a path of length $> k$, and Q_2 as the problem to decide whether G has a path of length $\leqslant k$. Then $\text{Accept}(Q_1 \vee Q_2)$ is the set of all graphs (actually all (G, k) pairs), whereas $\text{Accept}(Q_1 \wedge Q_2)$ is $\emptyset$. Both these problems are trivially in P.

5. Suppose that there exists an algorithm that, given a Boolean formula $\phi(x_1, x_2, \ldots, x_n)$ in $n \geqslant 1$ variables, decides in polynomial time whether $\phi(0, x_2, x_3, \ldots, x_n)$ is satisfiable. Which of the following statements is **not** true?

- **(A)** SAT can be solved in polynomial time
- **(B)** SAT is not NP-Complete
- **(C)** P = NP
- **(D)** NP = coNP

Solution Let $\phi' = \phi(\bar{x}_1, x_2, x_3, \ldots, x_n)$. Call the given algorithm twice, once on ϕ, and a second time on ϕ'. This gives a polynomial-time algorithm for SAT.

SAT is already proved to be NP-Complete. That proof does not become invalid in any case. Indeed, if P = NP, then every problem in this (common) class is NP-Complete.

6. Under the assumption that P ≠ NP, which of the following polynomial-time reductions is **not** possible?

- **(A)** CNFSAT $\leqslant$ DNFSAT
- **(B)** DNFSAT $\leqslant$ CNFSAT
- **(C)** SORTING $\leqslant$ VORONOI-DIAGRAM
- **(D)** HAM-PATH $\leqslant$ HALTING-PROBLEM

Solution CNFSAT is NP-Complete, whereas DNFSAT is in P, so a reduction CNFSAT $\leqslant$ DNFSAT implies DNFSAT is NP-complete, and so P = NP. (B) is true because CNFSAT is NP-Complete. (C) is true because we have seen reductions SORTING $\leqslant$CONVEX-HULL and CONVEX-HULL $\leqslant$ VORONOI-DIAGRAM. Since SAT is NP-Complete, we have HAM-PATH $\leqslant$ SAT, and we have also seen a reduction SAT $\leqslant$ HALTING-PROBLEM, so (D) is true.

7. Let NOTCONSTANT denote the problem of deciding whether a given Boolean formula is neither a tautology nor a contradiction. Under the assumptions that P ≠ NP and NP ≠ coNP, which of the following statements is true?

- **(A)** NOTCONSTANT is in P
- **(B)** NOTCONSTANT is not in NP
- **(C)** NOTCONSTANT is not in coNP
- **(D)** NOTCONSTANT is in NP ∩ coNP

Solution Work out that NONCONSTANT is NP-Complete. In (C) and (D), use the result that if any NP-Complete problem is in coNP, then NP = coNP.

8. Let SAME denote the problem of deciding whether two Boolean formulas ϕ_1 and ϕ_2 in the same variables evaluate to the same value for each truth assignment of the variables. Under the assumptions that P ≠ NP and NP ≠ coNP, which of the following statements is true?

- **(A)** SAME is in P
- **(B)** SAME is in NP but is not NP-complete
- **(C)** SAME is NP-Complete
- **(D)** SAME is coNP-complete

Solution Clearly, SAME is in coNP. Then use the reduction TAUTOLOGY $\leqslant$ SAME taking $\phi(x_1, x_2, \ldots, x_n)$ to $(\phi, (x_1 \vee \bar{x}_1) \wedge (x_2 \vee \bar{x}_2) \wedge \cdots \wedge (x_n \vee \bar{x}_n))$. Also use the fact if any coNP-Complete problem is in NP, then NP = coNP.

9. Let G be an undirected graph with n vertices. Under the assumption that P ≠ NP, which of the following problems is NP-Complete?

- **(A)** Decide whether G contains a clique of size $\leqslant n/5$.
- **(B)** Decide whether G contains a clique of size $\geqslant n/5$.
- **(C)** Decide whether G contains a clique of size $\geqslant n-5$.
- **(D)** Decide whether G contains a clique of size $\geqslant 5$.

Solution (A): Every graph contains a 1-clique.

(B): Reduction from CLIQUE. Let (G,k) be an instance of CLIQUE with $|V(G)| = n$. If $n \leqslant 5k$, add $5k - n$ isolated vertices. If $n > 5k$, add $\frac{n-5k}{4}$ vertices, and connect the new vertices to one another and to all of the n vertices of G.

(C): Exhaustively search over all subsets of V of size $n-5$. There are $\binom{n}{n-5} = \binom{n}{5}$ of them.

(D): Exhaustively search over all subsets of V of size 5.

In this exercise, the best strategy is to find the correct answer by elimination.

10. Let EVC denote the problem of deciding whether an undirected graph $G = (V, E)$ has a vertex cover U with $|U|$ even. Under the assumptions that $P \neq NP$ and $NP \neq coNP$, which of the following statements is true?

(A) EVC is in P
(B) EVC is NP-complete
(C) EVC is coNP-complete
(D) EVC is neither in NP nor in coNP

Solution Let $G = (V, E)$. If $|V|$ is even, take $U = V$. Otherwise take $U = V \setminus \{u\}$ for any $u \in V$.