CS101 Algorithms and Data Structures
Fall 2022
Homework 12

Due date: 23:59, December 18th, 2022

1. Please write your solutions in English.

2. Submit your solutions to Gradescope.

3. If you want to submit a handwritten version, scan it clearly.

4. When submitting, match your solutions to the problems correctly.

5. No late submission will be accepted.

6. Violations to any of the above may result in zero credits.

7. You are recommended to finish the algorithm design part of this homework with LaTeX.

8. Please check your Account Settings for Gradescope when submitting! Set your FULL name to your Chinese name and your 10-digit STUDENT ID correctly.

1. (0 points) Proving a problem in NP-Complete

When proving problem $A \in$ NP-Complete, please clearly divide your answer into the following (1+2) sections:

1. Show that $A \in$ NP.

2. Choose a problem $B \in$ NP-Complete and show that $B \leq_p A$.

    (a) For every instance I_B of B, construct an instance I_A of problem A.

    (b) Prove that $I_B \in B \iff I_A \in A$.

3. Conclude that A is in NP-Complete.

Proof Example

Suppose you are going to schedule courses for the SIST and try to make the number of conflicts no more than K. You are given 3 sets of inputs: $C = \{\cdots\}, S = \{\cdots\}, R = \{\{\cdots\}, \{\cdots\}, \cdots\}$. C is the set of distinct courses. S is the set of available time slots for all the courses. R is the set of requests from students, consisting of a number of subsets, each of which specifies the course a student wants to take. A conflict occurs when two courses are scheduled at the same slot even though a student requests both of them. Prove this schedule problem is NP-complete.

1. For any given schedule as a certificate, we can traverse every student's requests and check whether the courses in his/her requests conflicts and count the number of conflicts, and at last check if the total number is fewer than K, which can be done in polynomial time. Thus the given problem is in NP.

2. We show that 3-COLOR can be reduced to this problem.

    (a) For any instance of 3-coloring problem with graph G, we can construct an instance of the given problem: let every node v becomes a course, thus construct C; let every edge (u, v) becomes a student whose requests is $\{u, v\}$, thus construct R; let each color we use becomes a slot, thus construct S; at last let K equals to 0.

    (b) We now prove G is a yes-instance of 3-coloring problem if and only if (C, S, R, K) is a yes-instance of the given problem:

    • "⇒": if G is a yes-instance of 3-coloring problem, then schedule the courses according to their color. Since for each edge (u, v), u and v will be painted with different color, then for each student, his/her requests will not be scheduled to the same slot, which means the given problem is also a yes-instance.

    • "⇐": if (C, S, R, K) is a yes-instance of the given problem, then painting the nodes in G according to their slots. Since $K = 0$, then for every student, there is no conflict between their requests, which suggests that for every edge (u, v), u and v will not be painted with the same color. It is also a yes-instance of 3-coloring problem.

3. The given problem is in NP and a NP-Complete problem can be reduced to it in polynomial time, so it is also NP-Complete.

2. (6 points) Multiple Choices

Each question has **one or more** correct answer(s). Select all the correct answer(s). For each question, you will get 0 points if you select one or more wrong answers, but you will get 1 point if you select a non-empty subset of the correct answers.

Write your answers in the following table.

(a)	(b)	(c)
D	**AC**	**C**

(a) (2') A problem in NP is NP-Complete if:

    A. It can be reduced to another NP-Complete problem in polynomial time.

    B. There exists a NP-Complete problem which can be reduced to it in polynomial time.

    C. It can be reduced to any other NP problem in polynomial time.

    D. Any other NP problem can be reduced to it in polynomial time.

(b) (2') Assuming that $P \neq NP$, which of the following problems are in NP-Complete? You may search the Internet for more information if you are unfamiliar with the problems.

    A. `LONG-PATH`: (G, s, t, k) Given an undirected graph G, determine whether there exists a simple path from s to t whose length is greater or equal to k.

    B. `HALTING`: (P, I) Given a compilable C++ program P and the input I for P, determine if P runs infinitely on I.

    C. `4-SAT`: ϕ Given a CNF (conjunction normal form) where each clause is the disjunction of exactly 4 literals, determine whether ϕ is satisfiable.

    D. `PRIME`: n Given a positive integer n, determine whether it is a prime number.

(c) (2') For two decision problems A and B, suppose that $A \leq_p B$. Which of the following statements are true? (Hint: there exists complexity classes that are strictly bigger than NP)

    A. $A \in P \implies B \in P$

    B. $A \in \text{NP-Complete} \implies B \notin \text{NP-Complete}$.

    C. $B \in P \implies A \in P$.

    D. $B \in \text{NP-Complete} \implies A \in \text{NP-Complete}$.

3. **(10 points) PARTITION is** NP-Complete

   Given an array $A = [a_1, a_2, ..., a_n]$ of non-negative integers, consider the following problems:

   1.Partition: Determine whether there is a subset $P \subseteq [n]([n] = \{1, 2, ..., n\})$ such that $\sum_{i \in P} a_i = \sum_{j \in [n] \setminus P} a_j$.
   For example, given $A = [2, 4, 6, 8]$, then $P = \{1, 4\}$ is a partition of A since $a_1 + a_4 = a_2 + a_3 = 10$.

   2.Subset Sum: Given some integer K, determine whether there is a subset $P \subseteq [n]$ such that $\sum_{i \in P} a_i = K$.
   For example, given $A = [1, 3, 5, 7]$ and $K = 6$, then $P = \{1, 3\}$ gives a subset sum of $a_1 + a_3 = 6$.

   Suppose we have proven that Subset Sum problem is in NP-complete, prove that Partition problem is also in NP-complete.

   (a) (2') Prove that the partition problem is in NP.

   > **Solution:**
   > **Let (A,A') be the certificate. We can verify the certificate in linear time by summing the elements in each set, and comparing the result.**

   (b) (7') Find a polynomial time reduction from subset sum problem to parittion problem, and prove its correctness.

   > **Solution:**
   > Note that S' can be constructed in O(ISI) by first summing the elements in S and then adding {s−2t} to S, which proves the reduction is performed in polynomial time.
   > **∃T⟹ the set-partition problem on S' has a solution**
   > Suppose T exists. Let O be S - T. Let o be the sum of O. Let T'=T∪s−2t and t' be the sum of T'. make the following observations:
   > o = s - t
   > o - t = s - 2t   Difference in sum between O and T
   > t' = t + (s - 2t)
   >   = s - t   the sums of T'and O are equal
   >   = o
   > This proves:∃T⟹(T',O) is a solution to the set-partition problem on S'.
   > **The set-partition problem on S' has a solution ⟹∃T**
   > Suppose an equal-sum partitioning (A, A') of S'=S∪{s−2t} exists. The sum a of each partition must then be (s+(s -2t))/2=s-t. Consider the partition containing the element {s−2t} to be A'.Let A=A'−{s−2t}. We can observe S'−S={s−2t}⟹A⊆S. The sum of A is s-t-(s-2t)=t. In other words, A is a subset of S with sum t.

   (c) (1') Conclusion:

   > **Solution:**
   > **Partition is NP-Complete.**

4. (10 points) HALF-CLIQUE $\in$ NP-Complete

In an undirected graph $G = (V, E)$, a subset of the vertices $S \subseteq V$ is said to be a **clique** if for all pairs of vertices in S are connected or formally $\forall (u, v) \in S \times S(u \neq v \rightarrow \{u, v\} \in E)$. Note that a subset of zero or one vertex is also considered as a clique.

The k-CLIQUE problem is a classic NP-Complete problem stated as follows:
Given (k, G) where k is a non-negative integer and G is an undirected graph, determine if G contains a clique of at least k vertices.

Now let's consider the HALF-CLIQUE problem which is defined as follows:
Given an undirected graph $G = (V, E)$, determine if G contains a clique of $\lfloor V/2 \rfloor$ vertices. Show that HALF-CLIQUE is a NP-Complete problem.

Hint: reduce from k-CLIQUE, consider the cases where $k = |V|/2$, $k < |V|/2$ and $k > |V|/2$.

(a) (2') HALF-CLIQUE $\in$ NP

> **Solution:**
>
> The certificate is a subset V' of the vertices, which comprises the vertices belonging to the half-clique. We can validate this solution by checking that each pair of vertices belonging to the solution are adjacent, by simply verifying that they share an edge with each other. This can be done in polynomial time, that is O(V +E) using the following strategy of graph G(V,E).

(b) (7') HALF-CLIQUE $\in$ NP-Hard by providing k-CLIQUE $\leq_p$ HALF-CLIQUE

> **Solution:**
>
> **Let m be the number of nodes in the graph G.**
>
> **1 If k > = m/2, then for a constant number t, we add t nodes each of degree 0, for a graph G'. The graph G' has a total number of nodes equivalent to n = m + t, that is, the summation of all the nodes of graph G along with the extra nodes, such that it is equivalent to 2k, for any arbitrary value of k. Now k = n/2. This can be done by taking t = 2k -m. Then, the graph G has a clique of size k if and only if the graph G' has a clique of size k.**
>
> **2 If k < m/2 , then we add t additional nodes for the creation of graph G'. Edges can also be added from each new node to every other node in the graph. Therefore, any k-clique in G, for any arbitrary value of k combines with the t new nodes to make a (k+t)-clique in G', since edges have been added between each pair of vertices. A k+t-sized clique in G' must include at least k old nodes, which form a clique in the graph G. Therefore, the value of t is picked such that k+t = (m+t)/2, or t = m-2k, which makes the clique size in G' equivalent to n/2 exactly.**

(c) (1') Conclusion

> **Solution:** **HALF-CLIQUE is a NP-Complete problem**