CS101 Algorithms and Data Structures
Fall 2022
Homework 12
Reference Solution

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) Demand of the NP-complete Proof

When proving problem A is NP-complete, please clearly divide your answer into three steps:

1. Prove that problem A is in NP.
2. Choose an NP-complete problem B and for any B instance, construct an instance of problem A.
3. Prove that the yes/no answers to the two instances are the same.

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. Firstly, 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 choose 3-coloring problem which is a NP-complete problem. 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.
3. 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:
   - "$\Rightarrow$": 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.
   - "$\Leftarrow$": 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.

Therefore, the given problem is 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)
BD	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. $A \in \text{P} \implies B \in \text{P}$

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

C. $B \in \text{P} \implies A \in \text{P}$.

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: Simply iterate through P and $[n] \setminus P$ to verify that whether $\sum_{i \in P} a_i = \sum_{j \in [n] \setminus P} a_j$. This process has a time complexity of $\Theta(n)$

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

Solution: We can find a linear time reduction from Subset Problem, which is proved NP-complete, to Partition Problem.
Suppose we are given some A with target sum K. Let S be the sum of all elements in A.
If $S - 2K \geq 0$, generate a new set $A_0 = A \cup \{S - 2K\}$. If A_0 can be partitioned, then there is a subset of A that sums to K. We know that the two sets in our partition must each sum to $S - K$ since the sum of all elements will be $2S - 2K$. One of these sets, must contain the element $S - 2K$. Thus the remaining elements in this set sum to K.
If $S - 2K \leq 0$, generate a new set $A_0 = A \cup \{2K - S\}$. If A_0 can be partitioned, then there is a subset of A that sums to K. We know that the two sets in our partition must each sum to t since the sum of all elements will be $2K$. The set that does not contain $2K - S$ will be our solution to subset sum.
This reduction also clearly operates in $O(n)$, as we simply need to generate a new set with a single additional element (whose value is determined by summing all the elements of A).

(c) (1') Conclusion:

Solution: We have shown that partition problem is in NP, and partition problem $\leq_P$ subset sum problem, so partition problem is in 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: We show that HALF-CLIQUE is in NP by showing that YES instances of HALF-CLIQUE can be verified in polynomial time. A certificate can be a clique of the graph

Given an undirected graph $G = (V, E)$ and a subset S of the vertices V, we can verify if G is a YES instance of HALF-CLIQUE in the following procedure:

1. Check if $|S| = \lfloor |V|/2 \rfloor$.
2. Check if $S \subseteq V$.
3. For every distinct pair of vertices u, v in S, check if the edge $\{u, v\} \in E$
4. Answer YES if the input passes all the above testings. Otherwise, answer NO.

This can be done in polynomial time.

- If $G = (V, E)$ is in HALF-CLIQUE, then there exists $S \subseteq V$ such that $|S| = \lfloor |V|/2 \rfloor$ and S is a clique, we can use that set a certificate.
- If $G = (V, E)$ is not in HALF-CLIQUE, then for whatever set S, (G, S) cannot satisfy all the three conditions.

Thus, HALF-CLIQUE can be verified in polynomial time, so HALF-CLIQUE $\in$ NP.

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

Solution:
For every instance of the k-CLIQUE problem (k, G) where $G = (V, E)$ is an undirected graph, we can construct a graph G' in polynomial time such that $(k, G) \in$

k-CLIQUE $\iff$ $G' \in$ HALF-CLIQUE:

1. if $k = \lfloor |V|/2 \rfloor$, then let $G' = G$.
2. if $k < \lfloor |V|/2 \rfloor$, then obtain G' by adding $j = |V| - 2k$ vertices to the graph and connect them to every vertex in V.
3. if $k > \lfloor |V|/2 \rfloor$, then obtain G' by adding $j = 2k - |V|$ isolated vertices.

- $G \in k - \implies$ -CLIQUE G' HALF-CLIQUE: Let S be the k-clique in G.
  1. if $k = \lfloor |V|/2 \rfloor$, S is also a clique in G', and the size is $k = \lfloor |V|/2 \rfloor$.
  2. if $k < \lfloor |V|/2 \rfloor$, S and the j additional vertices can form a clique of size $k + j = |V| - k = \frac{|V|-2k+|V|}{2}$.
  3. if $k > \lfloor |V|/2 \rfloor$, S is also a clique in G', and the size is $k = \frac{|V|+j}{2} = \frac{2k-|V|+|V|}{2} = k$.

  Thus G' has a half-clique.
- $G' \in$ HALF-CLIQUE $\implies$ $G \in$ k-CLIQUE
  Let S' be the half clique in G', remove all the vertices that do not present in G to obtain another clique S.
  We claim that S is a clique in G of size at least k
  1. if $k = \lfloor |V|/2 \rfloor$, $S = S'$ is a clique in G of size is $k = \lfloor |V|/2 \rfloor$.
  2. if $k < \lfloor |V|/2 \rfloor$, S' is of size $\frac{|V|+|V|-2k}{2} = |V| - k$ thus the size of $|S| \geq |S'| - j = (|V| - k) - (|V| - 2k) = k$.
  3. if $k > \lfloor |V|/2 \rfloor$, S' must not contain any isolated vertices so $S' \subseteq V$ thus $S = S'$ is a clique in G of size k.

  Thus G contains a clique of at least k vertices. $G \in$ k-CLIQUE.

Thus, we have established polynomial time reudction from k-CLIQUE to HALF-CLIQUE.

(c) (1') Conclusion

Solution: We have shown that HALF-CLIQUE $\in$ NP and k-CLIQUE $\leq_p$ HALF-CLIQUE where k-CLIQUE $\in$ NP-Complete so HALF-CLIQUE $\in$ NP-Complete.