Algorithm Design and Analysis (Fall 2023)

Assignment 6

Deadline: Jan 9, 2023

Yuqing Tu 522030910152

Choose **two** of the first four questions to submit. Question 5 is the bonus question.

1. Prove that the following problem is NP-complete. Given an undirected graph G and an undirected graph H, decide if H is a subgraph of G.

2. Prove that the following problem is NP-complete. Given an undirected graph G and a positive integer $k \geq 2$, decide if G contains a spanning tree with maximum degree at most k.

3. Given an undirected graph $G = (V, E)$, prove that it is NP-complete to decide if G contains an independent set with size *exactly* $|V|/3$.

   Solution:

   1. Prove $|V|/3$ IndependentSet is in NP:
      To verify whether a vertices set S is an independent set with size $|V|/3$, we just need to verify whether $|S| = |V|/3$, whose time complexity is $O(|V|)$, and traverse E to verify whether there are some edge $e_i \in E$ that e_i's two endpoints are both in S, whose time complexity is $O(|E|)$ if we use a hash table to decide whether the endpoints is in S. So the total time complexity is $O(|V| + |E|)$. so $|V|/3$ IndependentSet is in NP.

   2. Prove $|V|/3$ IndependentSet is NP-hard:
      - In 3-CNF formula, for each clause, construct a triangle where there vertices represent there literals. Connect two vertices if one represents the negation of the other. Then we get a graph G.
        This reduction's time complexity is $O(|V| + |E|)$ because we just put each vertex and edge into G.
      - "yes to yes": If this 3SAT has a solution, then each clause has at least one literal which is `true`. For each triangle, put exactly one vertex representing a `true` literal into S. Then S is an independent set of G and $|S| = |V|/3$ because in each triangle, one and only one vertex is in S.
      - "no to no": Prove by contradiction. Assume that this 3SAT has no solution, but there is a vertices set S that is G's independent set and $|S| = |V|/3$. There are $|V|/3$ triangles in G totally, and in each triangle, one and only one vertex is in S. Otherwise there is at least one edge between some vertices in S. Then we assign `true` to all the literals representing the vertices in S. So each clause has one `true` literal, which means all clauses are `true`. Therefore, this 3SAT has a solution. There is a contradiction. "no to no" is true.

      Because 3SAT is NP-complete and 3SAT $\leq_k$ $|V|/3$ IndependentSet, so $|V|/3$ IndependentSet is NP-hard.

   To sum up, $|V|/3$ IndependentSet is NP-complete.

4. Consider the decision version of *Knapsack*. Given a set of n items with weights $w_1, \ldots, w_n \in \mathbb{Z}^+$ and values $v_1, \ldots, v_n \in \mathbb{Z}^+$, a capacity constraint $C \in \mathbb{Z}^+$, and a positive integer $V \in \mathbb{Z}^+$, decide if there exists a subset of items with total weight at most C and total value at least V. Prove that this decision version of Knapsack is NP-complete.

Solution:

1. Prove Knapsack is in NP:
   To verify whether a integer set S is a solution, we just need to compute $\sum_{i\in S} w_i$ and $\sum_{i\in S} v_i$. If $\sum_{i\in S} w_i \leq C$ and $\sum_{i\in S} v_i \geq V$, S is a solution, else S is not. And the time complexity is obviously $O(n)$. So Knapsack is in NP.

2. Prove Knapsack is HP-hard:
   - In a VectorSubsetSum problem, we construct n 2-dimensional vectors $x_i = (v_i, w_i)$ for $i = 1, 2, \ldots, n$. We also construct enough 2-dimensional vectors $x_{n+2i-1} = (v_{n+2i-1}, w_{n+2i-1}) = (-1, 0)$ and $x_{n+2i} = (v_{n+2i}, w_{n+2i}) = (0, 1)$ for $i = 1, 2, \ldots$. Then we decide if there is a integer set S making $\sum_{i\in S} x_i = (\sum_{i\in S} v_i, \sum_{i\in S} w_i) = (V, C)$.
     This reduction's time complexity is $O(n)$.
   - "yes to yes": If there is a integer set S making $\sum_{i\in S} x_i = (\sum_{i\in S} v_i, \sum_{i\in S} w_i) = (V, C)$, then we have $\sum_{i\in S, 1\leq i\leq n} v_i \geq V$ and $\sum_{i\in S, 1\leq i\leq n} w_i \leq C$, which is a solution of Knapsack.
   - "no to no": Prove by contradiction. Assume that there is not a integer set S' making $\sum_{i\in S'} x_i = (\sum_{i\in S'} v_i, \sum_{i\in S'} w_i) = (V, C)$, but Knapsack has a solution set S. Then we have $\sum_{i\in S} v_i = V' \geq V$ and $\sum_{i\in S} w_i = C' \leq C$. But

     $$\begin{aligned}
     &\sum_{i\in S} x_i + \sum_{1\leq i\leq V'-V} x_{n+2i-1} + \sum_{1\leq i\leq C-C'} x_{n+2i} \\
     &= (\sum_{i\in S} v_i - (V' - V), \sum_{i\in S} w_i + (C - C')) \\
     &= (V' - (V' - V), C' + (C - C')) \\
     &= (V, C)
     \end{aligned}$$

     So there is a set $S' = S + n + 2i - 1, n + 2j$ for $i = 1, 2, \ldots, V' - V$ and $j = 1, 2, \ldots, C - C'$ making $\sum_{i\in S'} x_i = (\sum_{i\in S'} v_i, \sum_{i\in S'} w_i) = (V, C)$. There is a contradiction. "no to no" is true.

   Because VectorSubsetSum is NP-complete and VectorSubsetSum $\leq_k$ Knapsack, so Knapsack is NP-hard.

To sum up, the decision version of Knapsack is NP-complete.

5. (**Bonus**) In the class, we have seen that 3SAT is NP-complete. In this question, we investigate the 2SAT problem and its variants. Similar to the 3SAT problem, in the 2SAT problem, we are given a 2-CNF Boolean formula (where each clause contains two literals) and we are to decide if this formula is satisfiable.

   (a) Prove that 2SAT is in P. (Hint: a clause $(a_i \lor a_j)$ with two literals a_i and a_j can be represented as two logical implications: $\neg a_i \implies a_j$ and $\neg a_j \implies a_i$; you may want to construct a directed graph with $2n$ vertices corresponding to $x_1, \neg x_1, x_2, \neg x_2, \dots, x_n, \neg x_n$.)

   (b) Consider this variant of the 2SAT problem: given a 2-CNF Boolean formula ϕ and a positive integer k, decide if there is a Boolean assignment to the variables such that at least k clauses of ϕ are satisfied. Notice that 2SAT is the special case of this problem with k equals to the number of the clauses. Prove that this problem is NP-complete.

Solution:

(a) Construct $2n$ vertices representing to all literals and their negation in the 2-CNF formula $x_1, \neg x_1, x_2, \neg x_2, \dots, x_n, \neg x_n$. And construct a directed edge from the vertex representing $\neg x_i$ to the vertex representing x_j and a directed edge from the vertex representing $\neg x_j$ to the vertex representing x_i if there is a clause $(x_i \lor x_j)$. Then we get a directed graph G. And this 2-CNF formula is not satisfiable iff there exists at least a pair of vertices representing x_i and $\neg x_i$ that can reach to each other.

Correctness proof:

Because $\neg a_i \implies a_j$ and $\neg a_j \implies a_i$ if there is a clause $(a_i \lor a_j)$, so if we start from one vertex representing x_i and assume x_i is `true`, then all literals represented by the vertices we can reach to are `true`.

- "if": Because there exists at least a pair of vertices representing x_i and $\neg x_i$ that can reach to each other, so x_i and $\neg x_i$ can't be `true` and `false` separately. Then this 2-CNF formula mustn't be satisfiable.
- "only if": Because this 2-CNF formula is not satisfiable, so there is at least a pair literals x_i and $\neg x_i$, if we want to make the formula `true`, x_i and $\neg x_i$ must be `true` at a same time. Then the pair of vertices representing x_i and $\neg x_i$ that can reach to each other.

Time complexity:

We just need to run DFS from each vertices x_i in G and decide whether we can reach to $\neg x_i$, so the total time complexity is $O(|V|(|V| + |E|)) = O(n^3)$ and it is a polynomial time algorithm.

So 2SAT is in P.

6. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.

   It takes me about 2 hours to finish the assignment. I give 2 to the difficulty.