CS3230
Tutorial 9

1. Consider the following problem:

   Input: Given a weighted graph G, two vertices u and v in G, and a value d.
   Question: Is there a path from u to v of weight at most d?

   Is the above problem in NP? Could it be NP-complete?

   Ans: Yes, it is in P (and thus in NP), as we can solve the problem using Dijkstra's algorithm in polynomial time, which gives shortest path from single source to any other node. Thus, given the shortest path from u to v, we can check if its weight is $\leq d$ or not.

   If $P = NP$, then it would be NP-complete. If P$\neq$ NP, then it will not be NP-complete.

2. In class we saw that it is open at present whether $P = NP$ or not. It is also open whether $NP = EXP$ or not. Is it possible that both $P = NP$ and $NP = EXP$ are true?

   Ans: No. As that would imply $P = EXP$, which is known not to be true, as mentioned in class.

3. Discrete knapsack problem is the knapsack problem we did in class where one has to either pick the whole item or none of it (i.e., we cannot pick a fraction of an item).

   It can be shown that discrete knapsack problem is NP-complete.

   Thus, if discrete knapsack problem can be solved in polynomial time, then all problems in NP can be solved in polynomial time.

   Professor S claimed that he could solve the discrete knapsack problem in time proportional to $C * n$ (see the dynamic programming algorithm done in class), where C is the capacity of the knapsack and n is the number of objects in the problem. Thus the discrete knapsack problem is in P.

   Thus, Professor S claimed that he has shown P=NP. Could you find a flaw in his argument?

   Ans: The complexity, as claimed by Professor S is not polynomial in terms of the input size: C can be coded using $\log C$ bits! Thus, the complexity of the algorithm is exponential in the input size.

4. Show that testing whether a graph $G = (V, E)$ is a subgraph of graph $G' = (V', E')$ is in NP.

   Ans: The intention of the problem was to consider subgraph when the "names" of vertices are not important.

Suppose $V = \{v_1, v_2, \ldots, v_n\}$ and $V' = \{v'_1, v'_2, \ldots, v'_r\}$.

1. Certificate would be a 1–1 mapping f from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, r\}$ which witnesses the subgraph property.

2. Verifier checks whether

for all i, j, $1 \leq i, j \leq n$, if $(i, j) \in E$ then $(f(i), f(j)) \in E'$,

If so, then it accepts (G is a subgraph of G'), otherwise it rejects.

Note that G is a subgraph of G' iff there exists such a mapping f.

5. A coloring of a graph $G = (V, E)$ is assignment of colors to each vertex of a graph such that if (u, v) is an edge, then the color assigned to u and v are different. A graph is k-colorable, if one can color the graph (with above constraint) using k colors.

   Show that checking whether a graph G is k-colorable is in NP.

   Ans:

   1. Certificate would be appropriate coloring of the graph using colors $1, 2, \ldots, k$. That is a mapping f from V to $\{1, 2, \ldots, k\}$.

   2. Verifier checks

   (a) if for all $v \in V$, $f(v) \in \{1, 2, \ldots, k\}$.

   (b) for all $(u, v) \in E$, $f(u) \neq f(v)$.

   This checking can be done in polynomial time.

6. In class you were told the decision problem regarding satisfiability (SAT).

   Consider the corresponding function problem,

   Input: A set of variables V and a set of clauses $C = \{c_1, c_2, \ldots, c_n\}$.

   Output: An assignement to the variables $v(x) \in \{true, false\}$, such that if the set of clauses is satisfiable, then the assignment $v(x)$ makes all the clauses true.

   Show how you could solve the above problem, in polynomial time, if you are given a "subroutine" (as black box) to solve the SAT problem in linear time.

   Ans: Suppose the variables are $x_1, x_2, \ldots, x_k$.

   $R =$ input CNF.
   Check using subroutine for SAT, whether R is satisfiable. If not, then quit (giving the answer, not satisfiable).
   For $i = 1$ to k {
   Let R' be obtained from R by substituting true for x_i
   Let R'' be obtained from R by substituting false for x_i
   Check using subroutine for SAT, whether R' is satisfiable.
   If so, then let $R = R'$, and assign $value(x_i) = true$.
   Else (R'' is satisfiable), let $R = R''$, and assign $value(x_i) = false$.
   }
   Output the truth assignment given by $value()$.

The above algorithm will either output not-satisfiable (if the original CNF formula is not satisfiable) or output an assignment $value()$ which makes the formula true.

7. Consider the following decision problem:

   Input: A set of variables V and a boolean formula F (which uses only variables from V).

   Question: Is there a truth assignment to the variables which makes the formula true?

(a) Is the above problem in NP?

(b) Is the above problem NP-complete? Give reasoning for your answer.

Ans: (a) Yes. Certificate would be assignment of truth values to the variables which makes the formula true. Verifier accepts iff the certificate is correct by checking if the truth assignment makes the formula true (note that this can be done in polynomial time).

(b) Yes, it is NP-complete, as it is in NP (part (a)), and also NP-hard, (as SAT problem is reducible to the above problem via identity function).