

CS375 Weekly Assignment 10

Due May 7th 10PM ET

A general note for CS375: When writing up your homework, please write neatly and explain your answers clearly, giving all details needed to make your answers easy to understand. Graders may not award full credit to incomplete or illegible solutions.

You are encouraged to discuss the problems with your classmates, **but your submission should be your own**. When writing your solutions up, you should not be looking at any other student's submission.

Clear communication **is the point**, on every assignment. In general in CS375, unless explicitly specified otherwise, answers should be accompanied by explanations. Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Necessary acknowledgements: many of these questions are reproductions or adaptations of questions of Eric Aaron's devising. Others are from our course texts as listed on the syllabus.

Recall the following problems and definitions from class:

SUBSETSUM

Input: A list of positive integers $x_1, \dots, x_n$ and a target T .

Output: TRUE if there exists a subset S of the x_i s such that $\sum_{x \text{ in } S} x = T$, otherwise FALSE.

VERTEXCOVER

Input: A graph G with parameter k .

Output: TRUE if there exists a subset S of k vertices such that every edge in the graph has an endpoint in S , otherwise FALSE.

3SAT

Input: A boolean formula that consists of the conjunction (AND, &&) of clauses, where each clause is the disjunction (OR, ||) of three boolean variables or their negations.

Output: TRUE if there exists an assignment of TRUE/FALSE to the boolean variables that makes the whole formula evaluate to TRUE, otherwise FALSE.

P = all problems that can be solved in polynomial time

NP = all problems whose solutions can be verified in polynomial time

NP-HARD = problems B that are so hard that $A \leq_p B$ for all A in **NP**

NP-COMPLETE = problems that are both **NP-HARD** and in **NP**

The Cook-Levin Theorem = 3SAT is **NP-HARD**

1. One might notice that I often loosely switch between the 'decision' version of problems and their optimization version. For example, with VERTEXCOVER, one could instead state an optimization version:

MINVERTEXCOVER

Input: A graph G

Output: A collection S of vertices of minimum size such that every edge in the graph has an endpoint in S

- (a) Suppose I have a polynomial time algorithm for MINVERTEXCOVER. Describe a polynomial time algorithm for VERTEXCOVER (**which takes a second parameter k**). This one should be easy.
 - (b) Suppose I have a polynomial time algorithm for VERTEXCOVER (**which takes a second parameter k**). Describe a polynomial time algorithm for MINVERTEXCOVER. This one is not as easy.
2. Find a student in an intro-ish CS Course (like 15X or 2XX) and explain to them what it means for a problem to be **NP-HARD**. Record what you said that got them to understand it. Obviously I have no way of verifying you did this, but fundamentally you don't understand something until you're capable of explaining it to someone else.
3. Let P be some problem that is **NP-COMPLETE**.
- (a) Explain why if P is in **P** then $\mathbf{P} = \mathbf{NP}$.
 - (b) Explain why if P is not in **P** then $\mathbf{P} \neq \mathbf{NP}$.
 - (c) Suppose B is some other problem and we're able to show that
 - B is in **NP**
 - $P \leq_p B$

Explain why this means that B is **NP-COMPLETE**.

4. Following the reduction from class from 3SAT to VERTEXCOVER and the following inputs for 3SAT:

$$\begin{aligned} F_1 : \quad & (x_1 || x_2 || x_3) \&\& (x_1 || x_2 || \bar{x}_3) \&\& (x_1 || \bar{x}_2 || x_3) \\ & \&\& (x_1 || \bar{x}_2 || \bar{x}_3) \&\& (\bar{x}_1 || x_2 || x_3) \&\& (\bar{x}_1 || x_2 || \bar{x}_3) \\ & \&\& (\bar{x}_1 || \bar{x}_2 || \bar{x}_3) \end{aligned}$$

$$\begin{aligned} F_2 : \quad & (x_1 || x_2 || x_3) \&\& (x_1 || x_2 || \bar{x}_3) \&\& (x_1 || \bar{x}_2 || x_3) \\ & \&\& (x_1 || \bar{x}_2 || \bar{x}_3) \&\& (\bar{x}_1 || x_2 || x_3) \&\& (\bar{x}_1 || x_2 || \bar{x}_3) \\ & \&\& (\bar{x}_1 || \bar{x}_2 || x_3) \&\& (\bar{x}_1 || \bar{x}_2 || \bar{x}_3) \end{aligned}$$

- (a) Find a satisfying assignment for F_1
- (b) Explain why no assignment for the boolean variables of F_2 will make the formula evaluate to TRUE
- (c) Draw the construction of the equivalent input for VERTEXCOVER for F_1 : in a different color (or any other signifying way), include the final triangle of vertices representing the clause that is present in F_2 that isn't present in F_1 .
- (d) Find a vertex cover of size $k = n + 2m = 3 + 2 * 7 = 17$ for the original graph (without the added clause).
- (e) Explain why you can't pick an additional two vertices from the last triangle (the added clause) that would give a vertex cover of the whole graph (with the added clause).