CSE 421 Winter 2026: Section 9

March 5, 2026

> **Instructions:** This section worksheet is designed to assist you in working through some example problems and developing your basics. You are encouraged to collaborate on the problems in this section worksheet as well as use the course's generative AI.

In this worksheet, we will practice polynomial-time reductions between classical NP-complete problems. Throughout, a reduction $A \leq_p B$ means that there exists a polynomial-time computable mapping from instances of A to instances of B such that YES instances map to YES instances and NO instances map to NO instances.

All the NP-complete we will use in this section sheet were introduced in last week's worksheet. Take a look at that sheet if you need reminders for the problem definitions.

1 Lecture Review: 3SAT $\leq_p$ Vertex Cover

In the lecture, we described the standard reduction from 3-SAT to Vertex Cover. The intention of the section is to go over the reduction once more, this time highlighting any concepts that you may have missed.

In order to prove the reduction, we need to identify a poly-time computable function f mapping instances of 3-SAT to instances of vertex cover with the property that

$$\varphi \text{ is a "yes" instance of 3-SAT} \iff f(\varphi) \text{ is a "yes" instance of Vertex Cover}.$$

Constructing such a reduction will involve mapping all the structures of the original formula φ so that they are incorporated in the pair $(G, k) = f(\varphi)$ that forms the vertex cover instance. In particular, we should ask and answer the following questions:

1. How is the structure of a clause $(y_i \vee y_j \vee y_k)$ represented in the graph G for any triple of literals?
2. How does the graph incorporate that a variable and its negation cannot both be set to be true in any satisfying assignment?
3. If x is a satisfying assignment of φ, how can we use x to find a vertex cover for (G, k)?
4. Secondly, if we find a vertex cover for (G, k), what structures must the vertex cover have? How can we use that to identify an assignment of vertices for

After having painted some picture of the reduction, go through and systematically answer the following questions to generate the reduction.

1. Describe the construction of the graph G given a 3SAT formula:
   - What vertices are created for each variable?
   - What vertices and edges are created for each clause?
2. What value of k is used?
3. Prove correctness:
   (a) (YES $\Rightarrow$ YES) If φ is satisfiable, show the graph has a vertex cover of size at most k.
   (b) (YES $\Leftarrow$ YES) If the graph has a vertex cover of size at most k, show φ is satisfiable.

2 Vertex Cover $\leq_p$ Independent Set

Let us define another problem: Independent Set. Like Vertex Cover, an instance of Independent Set consists of an undirected graph $G = (V, E)$ and an integer ℓ. It is a YES instance if there exists $I \subseteq V$ with $|I| \geq \ell$ such that no two vertices in I share an edge. It is a NO instance otherwise.

Vertex Cover and Independent Set are complementary problems to each other. Prove the reduction Vertex Cover $\leq_p$ Independent Set.

If stuck, pick a graph G and draw both the min vertex cover and max independent set. How do these objects relate?

3 Vertex Cover $\leq_p$ Subset Sum (Challenging)

We now prove Subset Sum is NP-complete by reducing Vertex Cover to Subset Sum. As we showed in class that Subset Sum reduces to Knapsack, this also proves that Knapsack is NP-complete. For this problem, we provide the reduction; you only need to prove that it is correct.

Given an instance (G, k) of vertex cover, assume there are n vertices of the graph enumerated $\{1, \dots, n\}$ and m edges in the graph with the edges enumerated $\{1, \dots, m\}$. Then construct "vertex integers" and "edge integers" as part of our reduction:

1. "Edge integers": For every edge $e \in \{1, \dots, m\}$, include the integer $a_e := 10^e$.
2. "Vertex integers": For every vertex $v \in \{1, \dots, n\}$, include the integer $b_v := 10^{m+1} + \sum_{e : e \ni v} 10^e$.

3. Let $T := k \cdot 10^{m+1} + \sum_{e=1}^{m} 2 \cdot 10^e$.

4. Output the Subset Sum instance defined by positive integers $a_1, \dots, a_m, b_1, \dots, b_n$ and target T.

Concrete Example. Consider a yes-instance of vertex cover with:

$$V = \{1,2,3\}, \quad E = \{e_1 = (1,2),\ e_2 = (2,3),\ e_3 = (1,3)\},\ k = 2\,.$$

Then, by construction, the "edge" integers are

$$a_1 = 10^1 = 10, \quad a_2 = 10^2 = 100, \quad a_3 = 10^3 = 1000.$$

And the "vertex" integers are

$$b_1 = 10^4 + 10^1 + 10^3 = 10000 + 10 + 1000 = 11010,$$

$$b_2 = 10^4 + 10^1 + 10^2 = 10000 + 10 + 100 = 10110,$$

$$b_3 = 10^4 + 10^2 + 10^3 = 10000 + 100 + 1000 = 11100.$$

Since $k = 2$ and $m = 3$,

$$T = 2 \cdot 10^4 + 2(10 + 100 + 1000) = 20000 + 2220 = 22220.$$

A valid vertex cover for the graph is selecting vertices 1 and 3. In this graph, edges e_1 and e_2 are covered by exactly one vertex, while the edge e_3 is covered by two vertices. Observe that

$$b_1 + b_3 + a_1 + a_2 = T\,. \tag{1}$$

1. **"Yes" → "Yes"**: Assume that G has a vertex cover consisting of vertices $v_1, \dots, v_{k'}$ for $k' \leq k$. Construct *any* subset of positive integers amongst $a_1, \dots, a_m, b_1, \dots, b_n$ that sum to the target T. Prove that your answer is correct.

2. **"Yes" ← "Yes"**: Assume there exists a subset of positive integers amongst $a_1, \dots, a_m$, $b_1, \dots, b_n$ that sum to the target T. Prove that

   (a) the subset must include exactly k "vertex integers" and

   (b) for every edge $e = (u, v)$, at least one of the integers b_u or b_v must be included in the subset.

   This proves that the corresponding vertices form a vertex cover.

Hint: When the integers a_e or b_v are written in base 10, they are expressed only using the digits 0 and 1. Then, argue digit-by-digit in base 10. Explain why no carries can occur.

Discussion Question

In the reduction above, we used powers of 10.

1. Why would using powers of 2 (i.e., binary encoding) not work cleanly?
2. More generally, what property must the base B satisfy so that no carries can occur?

4 Solutions

4.1 Lecture Review: 3SAT $\leq_p$ Vertex Cover

Reduction. Let φ be a 3CNF formula with clauses $C_1, \dots, C_m$, where $C_j = (\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ and each $\ell_{jt} \in \{x_i, \neg x_i\}$. Construct a graph G with one vertex for each *clause occurrence* of a literal: denote these vertices by v_{j1}, v_{j2}, v_{j3}, where v_{jt} is labeled by ℓ_{jt}. For each clause C_j, connect $\{v_{j1}, v_{j2}, v_{j3}\}$ into a triangle (add all three edges among them). Next, for any two vertices v_{jt} and $v_{j't'}$ labeled by complementary literals (one is x_i and the other is $\neg x_i$ for some i), add the edge $\{v_{jt}, v_{j't'}\}$. (Equivalently: for each variable x_i, make all occurrences of x_i adjacent to all occurrences of $\neg x_i$.) Set $k := 2m$ and output the Vertex Cover instance (G, k).

"yes" → "yes". Assume φ is satisfiable and fix a satisfying assignment α. We build a vertex cover S of size exactly $2m$ by taking *two* vertices from each clause triangle. For each clause C_j, choose one literal in the clause that is true under α, and let v_{jt^*} be the corresponding clause vertex. Put the other two vertices of that triangle into S, and leave $v_{jt^*} \notin S$. After doing this for every clause, we have selected exactly $2m$ vertices. This set covers every edge *within* a clause triangle since any edge of the triangle is incident to one of the two chosen vertices. It remains to show that this covers the edges added between complementary literals across clauses. Consider any such edge $\{u, v\}$ where u is labeled x_i and v is labeled $\neg x_i$. We claim it cannot happen that both endpoints are omitted from S. Indeed, an endpoint is omitted only when its label was chosen as the designated *true* literal for its clause. Thus omitting both endpoints would mean that α makes both x_i and $\neg x_i$ true, which is impossible. Therefore at least one endpoint of every cross-clause "complement" edge lies in S, so all edges are covered and $(G, 2m)$ is a YES instance of Vertex Cover.

"yes" ← "yes". Assume G has a vertex cover S with $|S| \leq 2m$. Each clause gadget is a triangle, and any vertex cover of a triangle has size at least 2, so S must contain at least $2m$ vertices total across the m disjoint triangles. Since $|S| \leq 2m$, it follows that $|S| = 2m$ and, moreover, S contains *exactly two* vertices from each clause triangle. For each clause C_j, let w_j be the unique vertex of that triangle that is *not* in S. Define an assignment α by setting each literal labeling w_j to be true (i.e., if w_j is labeled x_i, set $\alpha(x_i) = \text{true}$; if labeled $\neg x_i$, set $\alpha(x_i) = \text{false}$). This is well-defined: if for some variable x_i there were two clauses with uncovered vertices labeled x_i and $\neg x_i$, then those two uncovered vertices would be adjacent by construction (complementary-literal edge), but neither endpoint would be in S, contradicting that S covers that edge. Hence no variable is forced to be both true and false. Finally, every clause C_j is satisfied because w_j is one of its three literal-occurrence vertices and we set that literal to true. Therefore φ is satisfiable.

4.2 Vertex Cover $\leq_p$ Independent Set

Reduction. Given an instance (G, k) of Vertex Cover with $G = (V, E)$, output the Independent Set instance (G, ℓ) on the *same* graph G with $\ell := |V| - k$. This mapping is computable in polynomial time since it only computes $|V|$ and subtracts k.

"yes" → "yes". Assume (G, k) is a YES instance of Vertex Cover, so there exists a set $S \subseteq V$ with $|S| \leq k$ that hits every edge. Let $I := V \setminus S$. Then $|I| = |V| - |S| \geq |V| - k = \ell$. We claim I is an independent set. If not, then there exists an edge $\{u, v\} \in E$ with $u, v \in I$, meaning $u, v \notin S$. But then the edge $\{u, v\}$ is not covered by S, contradicting that S is a vertex cover. Hence I is an independent set of size at least ℓ, so (G, ℓ) is a YES instance of Independent Set.

"yes" ← "yes". Assume (G, ℓ) is a YES instance of Independent Set, so there exists an independent set $I \subseteq V$ with $|I| \geq \ell = |V| - k$. Let $S := V \setminus I$. Then $|S| = |V| - |I| \leq |V| - \ell = k$. We claim S is a vertex cover. Consider any edge $\{u, v\} \in E$. Since I is independent, it cannot contain both endpoints of an edge, so at least one of u or v lies outside I, i.e., in S. Thus every edge has an endpoint in S, so S is a vertex cover of size at most k. Therefore (G, k) is a YES instance of Vertex Cover.

4.3 Vertex Cover $\leq_p$ Subset Sum (Challenging)

Reduction. Given a Vertex Cover instance (G, k) where G has vertices $V = \{1, \dots, n\}$ and edges $E = \{1, \dots, m\}$ (with a fixed enumeration of edges), construct Subset Sum numbers as follows: for each edge $e \in \{1, \dots, m\}$ include an "edge integer" $a_e := 10^e$, and for each vertex $v \in \{1, \dots, n\}$ include a "vertex integer" $b_v := 10^{m+1} + \sum_{e:\, e \ni v} 10^e$. Set the target $T := k \cdot 10^{m+1} + \sum_{e=1}^{m} 2 \cdot 10^e$, and output the instance consisting of the multiset $\{a_1, \dots, a_m, b_1, \dots, b_n\}$ and target T. Intuitively, the $(m+1)$-st digit (the 10^{m+1} place) counts how many vertices we selected, and for each edge-position digit 10^e we want total digit-sum exactly 2, which we will achieve by contributing either $(1 + 1)$ from both endpoints via vertex integers, or $(1 + 0)$ from one endpoint plus an extra $(+1)$ via the edge integer a_e.

"yes" → "yes". Assume G has a vertex cover $S \subseteq V$ with $|S| \leq k$. If $|S| < k$, add arbitrary vertices to obtain a superset $S' \supseteq S$ with $|S'| = k$ (adding vertices preserves the vertex-cover property). We now choose a subset of integers summing to T as follows: include all vertex integers $\{b_v : v \in S'\}$, and additionally, for each edge $e = (u, v)$, include the edge integer a_e if and only if *exactly one* of $\{u, v\}$ lies in S' (i.e., the edge is covered by exactly one chosen endpoint). Consider the sum digit-by-digit in base 10. In the 10^{m+1} place, each chosen b_v contributes 1, so this digit sums to k, matching the $k \cdot 10^{m+1}$ term of T.

Now fix an edge position $e \in \{1, \dots, m\}$. Each selected vertex integer b_v contributes a 1 in the 10^e place exactly when v is incident to edge e. Thus the total contribution from selected vertex integers to digit e is the number of chosen endpoints of e, which is either 1 or 2 because S' is a vertex cover. If it is 2, we do not include a_e and the digit sum is 2. If it is 1, we include $a_e = 10^e$ and the digit sum becomes 2. Hence for every e, the 10^e digit sums to exactly 2, matching $\sum_{e=1}^{m} 2 \cdot 10^e$ in T. Finally, no carries occur because in every digit position $e \leq m$ the total is exactly $2 < 10$, and in the $m+1$ position the total is exactly k, which is still a single base-10 digit contribution with no interaction with lower digits (lower digits are < 10). Therefore the selected integers sum exactly to T, so the constructed Subset Sum instance is a YES instance.

"yes" ← "yes". Assume there exists a subset of the integers $\{a_1, \dots, a_m, b_1, \dots, b_n\}$ that sums to T. Write all numbers in base 10. Each a_e has a single 1 in digit position e and 0 elsewhere, while each b_v has a 1 in position $m+1$ and also 1's in the edge positions incident to v. The target T has digit k in position $m+1$ and digit 2 in each position $e \in \{1, \dots, m\}$, with all other digits 0.

We first argue that no carries can occur in any valid solution: the only way to affect digit $m+1$ is to select vertex integers, each contributing exactly 1 there, and the target at that digit is exactly k with no contribution from lower digits (since lower digits in T are $2 < 10$). If any carry into digit $m+1$ occurred from the 10^m place, it would require the 10^m digit-sum to be at least 10, but every summand contributes at most 1 to that digit, and the target digit is 2, so matching T forces that digit-sum to be exactly 2, hence < 10 and impossible to generate a carry. The same reasoning applies to every digit $e \leq m$: matching digit e to 2 forces the digit-sum to be exactly $2 < 10$, so no carry can occur anywhere. Consequently, we may reason digit-by-digit without carries.

From digit $m+1$, the sum equals k, so the subset contains exactly k vertex integers b_v (and any number of edge integers, which contribute 0 to that digit). Now fix an edge $e = (u, v)$ and look at digit e: each of b_u and b_v (if chosen) contributes 1 at digit e, and no other vertex integer contributes to digit e except endpoints of e; additionally, a_e (if chosen) contributes 1 at digit e. Since the total at digit e must be exactly 2 and carries do not exist, it is impossible that neither b_u nor b_v is chosen: in that case the only way to reach 2 would be to use a_e (giving at most 1) plus contributions from non-endpoints (which are 0), a contradiction. Therefore for every edge $e = (u, v)$, at least one of b_u or b_v is in the chosen subset. Let S be the set of vertices whose b_v are chosen; we have $|S| = k$ and every edge has an endpoint in S, so S is a vertex cover of size at most k. Hence (G, k) is a YES instance of Vertex Cover.