

Vertex Cover

- **Definition:** A vertex cover in a graph $G = (V, E)$ is a subset $C \subseteq V$ such that for every edge $\{u, v\} \in E$, either $u \in C$ or $v \in C$ (or both).
- **Definition:** VERTEX-COVER = $\{\langle G, k \rangle \mid G \text{ has a vertex cover of size } k \text{ or less}\}$
- **Theorem:** VERTEX-COVER is NP-complete.
- **Proof:**
 - First, we need to prove that VERTEX-COVER \in NP. The non-deterministic algorithm guesses a set of at most k vertices, and checks that the set is a vertex cover by verifying that it contains an endpoint of every edge in the graph.
 - Second, we need to prove that VERTEX-COVER is NP-hard. To prove this, we will show $3\text{SAT} \leq_p \text{VERTEX-COVER}$. The reduction maps a 3CNF formula φ to $\langle G, k \rangle$, where G is constructed as follows. The vertices of G are divided into two sets: “top” vertices and “bottom” vertices.
 - * For every variable x_i of φ , G contains two top vertices labelled x_i and \bar{x}_i , which are joined by an edge.
 - * For every clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$ of φ , G contains three bottom vertices labelled ℓ_1, ℓ_2 , and ℓ_3 , with edges between all three pairs.
 - * For every bottom vertex u and top vertex v with the same label, G contains the edge $\{u, v\}$.
 - The parameter k is set to $n + 2m$, where n is the number of variables in φ and m is the number of clauses. That is, the reduction maps φ to $\langle G, n + 2m \rangle$.
 - Suppose that φ is satisfiable. Let τ be a satisfying assignment for φ . Define a set $C \subseteq V$ as follows: for every pair of connected top vertices labelled x_i, \bar{x}_i , C contains the vertex labelled x_i if $\tau(x_i) = 1$, otherwise it contains the vertex labelled \bar{x}_i . For every triangle of bottom vertices labelled ℓ_1, ℓ_2, ℓ_3 , C contains two of these vertices: the vertex that is omitted from C is chosen so that its label is assigned the value 1 by τ (since τ satisfies φ there exists such a vertex). C contains $n + 2m$ vertices, so it remains to show that C is a vertex cover. Clearly C covers all the edges between top vertices, since it either contains the vertex labelled x_i or the vertex labelled \bar{x}_i ; and C covers all the edges between bottom vertices, since it contains two vertices from each triangle. Consider a particular edge $\{u, v\}$, where u is a top vertex, and v is a bottom vertex. By construction of G , u and v have the same label ℓ . If $v \notin C$, then by construction of C it follows that $\tau(\ell) = 1$, implying that $u \in C$. So the edges between top vertices and bottom vertices are also covered by C , and thus C is a vertex cover of size $n + 2m$.

- Now suppose that G has a vertex cover C of size $n + 2m$ or less. C must contain at least n top vertices, since it must include the vertex labelled x_i or the vertex labelled \bar{x}_i in order to cover the edge between these vertices. Similarly, C contains at least $2m$ bottom vertices, since it must take two vertices from each triangle to cover the edges within the triangle. It follows that $|C| = n + 2m$ and that C contains exactly n top vertices and exactly $2m$ bottom vertices. Thus for each variable x_i , C contains either the top vertex labelled x_i , or the top vertex labelled \bar{x}_i , but not both. Define a truth assignment τ : $\tau(x_i) = 1$ if C contains the top vertex labelled x_i , otherwise $\tau(x_i) = 0$. Then τ is a satisfying truth assignment for φ . To see this, consider any clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ of φ . C contains exactly two of the corresponding bottom vertices, so suppose it does not contain the vertex v labelled ℓ_1 . There is an edge between v and the top vertex u labelled ℓ_1 , and since C is a vertex cover it follows that $u \in C$. By construction of τ we have $\tau(\ell_1) = 1$, so τ satisfies C . Since τ satisfies every clause of φ , it is a satisfying assignment for φ .
- We have shown that $\langle \varphi \rangle \in 3\text{SAT} \iff \langle G, n+2m \rangle \in \text{VERTEX-COVER}$. The reduction can be carried out in polynomial time: G can be generated by iterating once through the clauses of φ and creating the appropriate subgraphs.

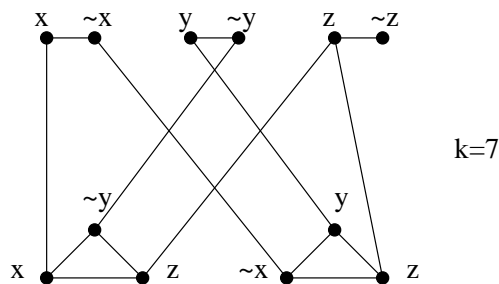


Figure 1: Graph corresponding to $(x \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee z)$