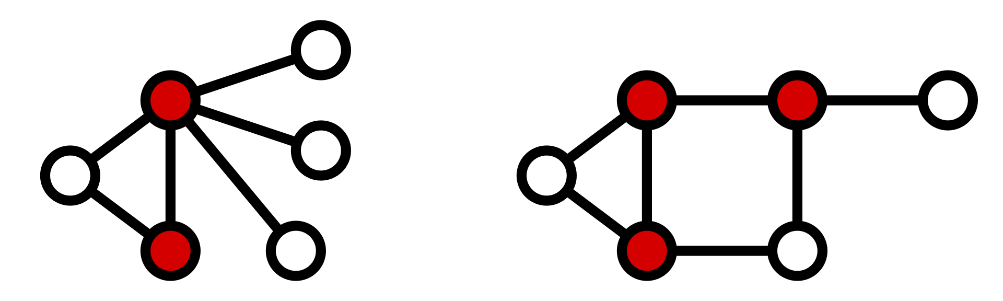
Vertex cover

**Definition**:

A vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.

Example:



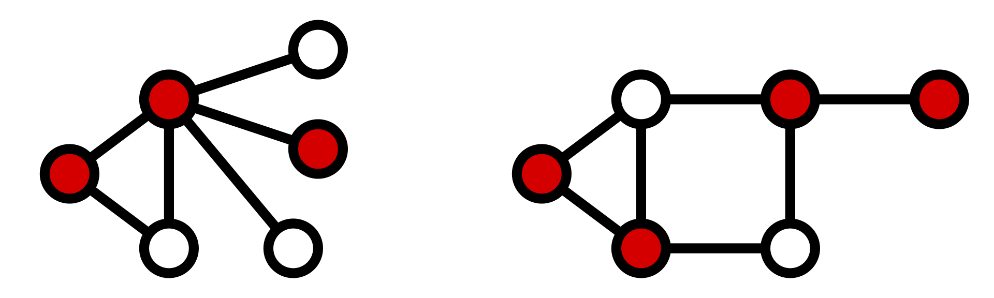
We want the size of vertex cover as small as possible, so we call a vertex cover of smallest possible size as a minimum vertex cover.

Let *V'* be the minimum vertex cover.

*τ* be the size of minimum vertex cover

⇒ {\displaystyle \tau =|V'|} |*V'*| = *τ*.

The following figure shows examples of minimum vertex cover in the previous graphs[1].



**Decision problem level**:

In decision version, we called Vertex Cover problem, which is a NP-complete problem.

Proof:

An instance of the Vertex Cover problem is a graph *G* (*V*, *E*) and a positive integer *k*, and the problem is to check whether a vertex cover of size at most *k* exists in *G*. By definition, an NP-complete problem is a problem which is both in NP and NP-hard, the proof consists of two parts:

1. **Proof that vertex cover is in NP**  
   If any problem is in NP, then given a solution to the problem, to verify whether the solution is correct or not in polynomial time.

The solution for the Vertex Cover problem is a subset *V'* of *V*, which contains the vertices in the vertex cover. We can check whether the set *V'* is a vertex cover of size *k* using the following strategy:

let *count* be an integer

set *count* to 0

for each vertex *v* in *V'*

remove all edges adjacent to *v* from set *E*

increment count by 1

if *count* = *k* and *E* is empty

then the given solution is correct

else

the given solution is wrong

It is clear that this can be done in polynomial time. Thus the Vertex Cover problem is in the class NP.

1. **Proof that vertex cover is NP Hard**   
   To prove that Vertex Cover is NP Hard, we take Clique problem that has already to be prove NP Hard, and show that Clique problem can be reduced to the Vertex Cover problem.

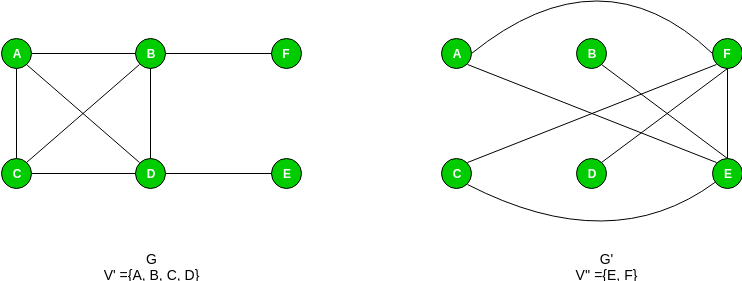
*“In computer science, the clique problem is the computational problem of finding cliques (subsets of vertices, all adjacent to each other, also called complete subgraphs) in a graph.”*

Here, an instance of the Clique problem is a graph *G* (*V*, *E*) and a non-negative integer *k*, the problem is finding out whether there is a clique of size *k* in the given graph.

⇒)

Now, we need to show that any instance (*G*, *k*) of the Clique problem can be reduced to an instance of the Vertex Cover problem. Let *G'* be the complement graph of *G*, which consists of all edges not in *G*, but all vertices in *G*. Now, the problem of finding whether a clique of size *k* exists in the graph *G* is the same as the problem of finding whether there is a vertex cover of size |*V*| – *k* in *G'*. We need to show that this is indeed the case.

To understand the proof, consider the following example graph and its complement:



Assume that there is a clique of size *k* in *G*. Let the set of vertices in the clique be *V'*, which means |*V'* | = *k*. In the complement graph *G'*, let *V''* be the vertex set of *G'*, when we pick any edge (*u*, *v*). Then at least one of *u* or *v* must be in *V''*. If not, both *u* and *v* were in *V'*, then the edge (u, v) would belong to *V'*, which, in turn would mean that the edge (u, v) is in *G*. (→← ∴since (u, v) is not in G) Thus, all edges in *G'* are covered by vertices in the set *V''.*

⇐)

Now assume that there is a vertex cover *V''* of size |*V*| – *k* in *G'*. This means that all edges in *G'* are connected to some vertex in *V''*. As a result, if we pick any edge (*u*, *v*) from *G'*, both vertices *u* and *v* cannot be outside the set *V''*. This means, all edges (*u*, *v*) where both u and v are outside the set *V''* are in *G*, i.e., these edges constitute a clique of size *k*.

Thus, we can say that there is a clique of size *k* in graph *G* if and only if there is a vertex cover of size |*V*| – *k* in *G'*, and hence, any instance of the Clique problem can be reduced to an instance of the Vertex Cover problem. Thus, Vertex Cover is NP Hard. Since vertex cover is in both NP and NP Hard classes, it is NP Complete[2].

**Exact evaluation**:

For bipartite graphs, the equivalence between vertex cover and maximum matching described by Kőnig's theorem allows the bipartite vertex cover problem to be solved in polynomial time.

For tree graphs, an algorithm finds a minimal vertex cover in polynomial time by finding the first leaf in the tree and adding its parent to the minimal vertex cover, then deleting the leaf and parent and all associated edges and continuing repeatedly until no edges remain in the tree.

**Pseudocode**:

APPROXIMATION-VERTEX-COVER(G)=

C = ∅

E'= G.E

**while** E' ≠ ∅: #如果邊集合不等於空集合

#讓(u, v)表示為集合E' 中任意一條邊

let (u, v) be an arbitrary edge of E'

#將{u, v} 兩點中，其中一點加入vertex cover C中

C = C ∪ {u, v}

#移除集合E' 中所有和u 或 v 相連的所有邊

remove from E' every edge incident on either u or v

**return** C

**Source**:

<https://en.wikipedia.org/wiki/Vertex_cover>

<https://www.geeksforgeeks.org/proof-that-vertex-cover-is-np-complete/>