DAA - Vertex Cover

A vertex-cover of an undirected graph ***G = (V, E)*** is a subset of vertices $V' \subseteq V$ such that if edge ***(u, v)*** is an edge of ***G***, then either ***u*** in ***V*** or ***v*** in V' or both.

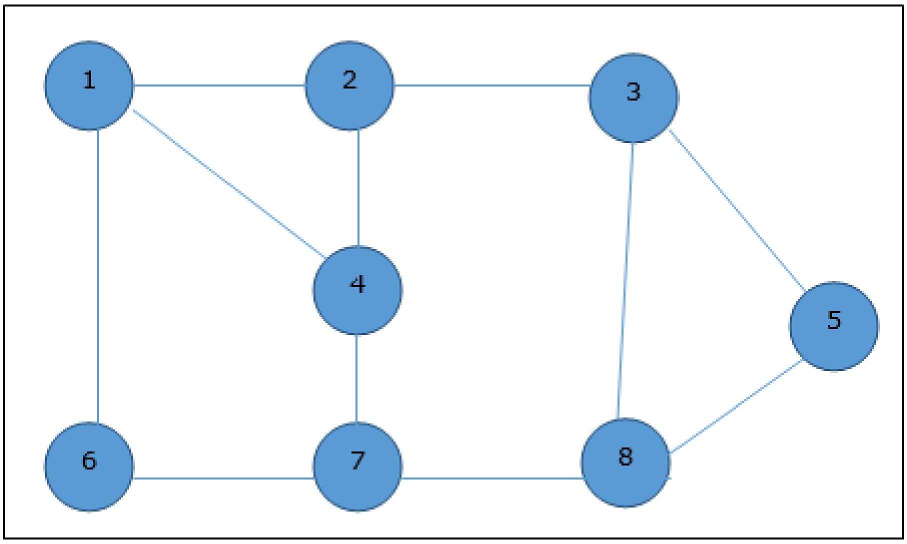
Find a vertex-cover of maximum size in a given undirected graph. This optimal vertexcover is the optimization version of an NP-complete problem. However, it is not too hard to find a vertex-cover that is near optimal.

```
APPROX-VERTEX_COVER (G: Graph) c ← { } E' ← E[G]
while E' is not empty do
   Let (u, v) be an arbitrary edge of E' c ← c U {u, v}
   Remove from E' every edge incident on either u or v
return c
```

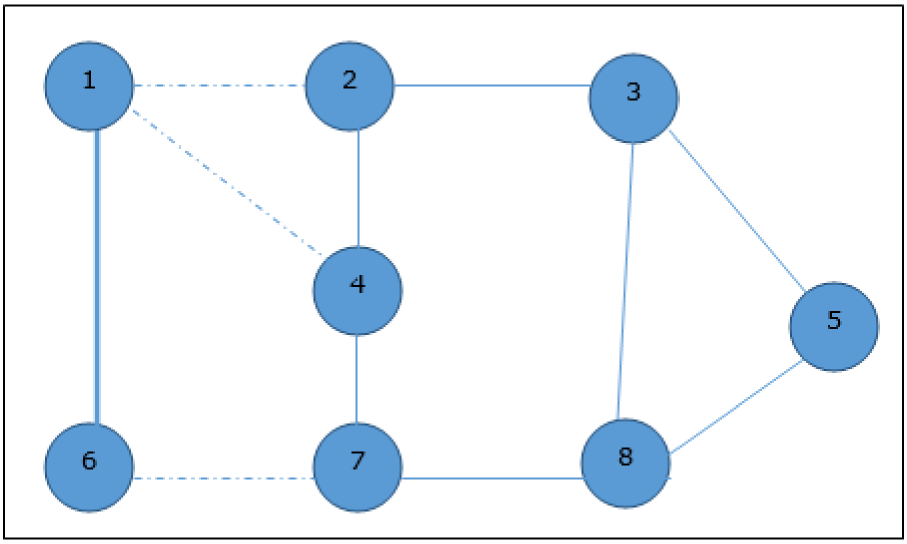
Example

The set of edges of the given graph is −

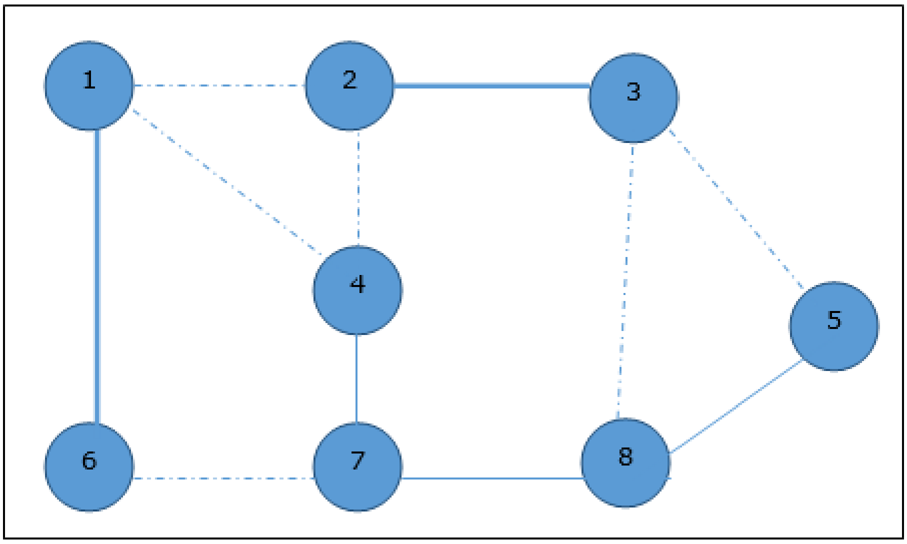
{(1,6),(1,2),(1,4),(2,3),(2,4),(6,7),(4,7),(7,8),(3,8),(3,5),(8,5)}



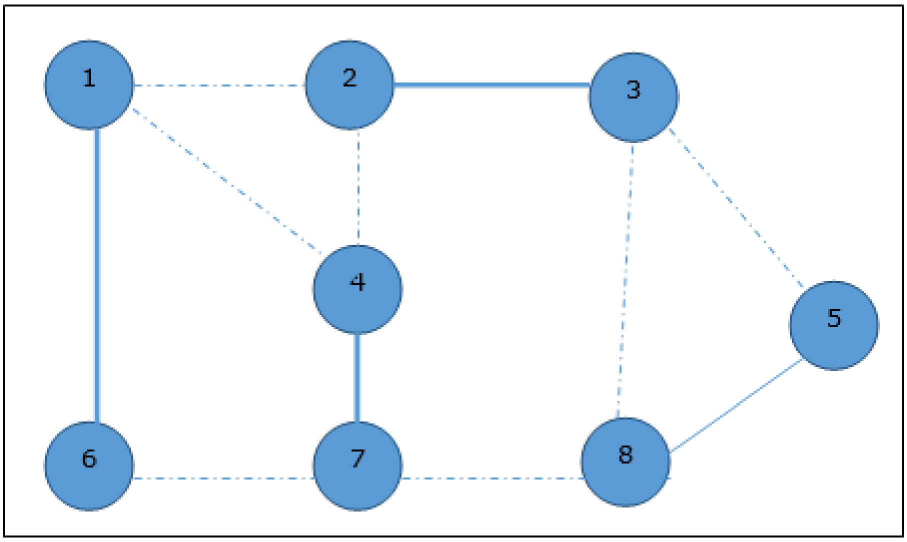
Now, we start by selecting an arbitrary edge (1,6). We eliminate all the edges, which are either incident to vertex 1 or 6 and we add edge (1,6) to cover.



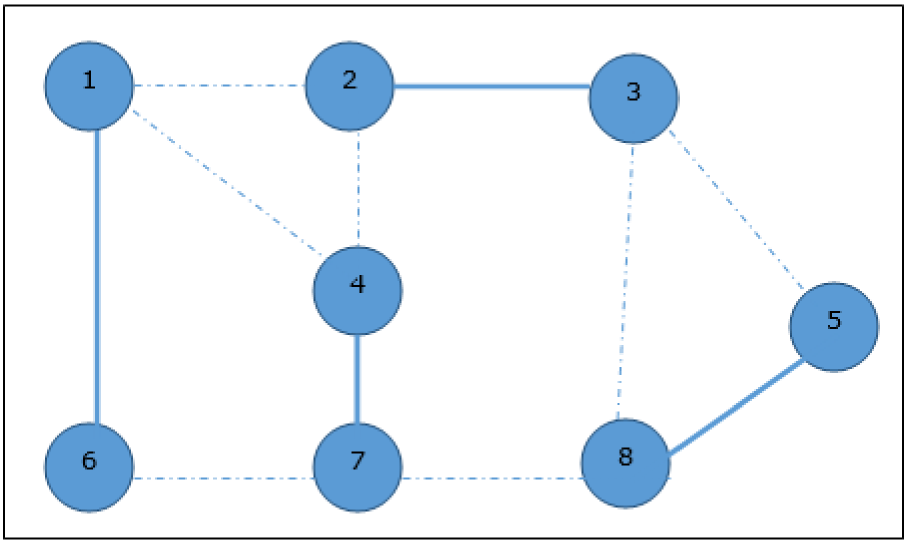
In the next step, we have chosen another edge (2,3) at random



Now we select another edge (4,7).



We select another edge (8,5).



Hence, the vertex cover of this graph is {1,2,4,5}.

Analysis

It is easy to see that the running time of this algorithm is $O(V + E)$, using adjacency list to represent E'.