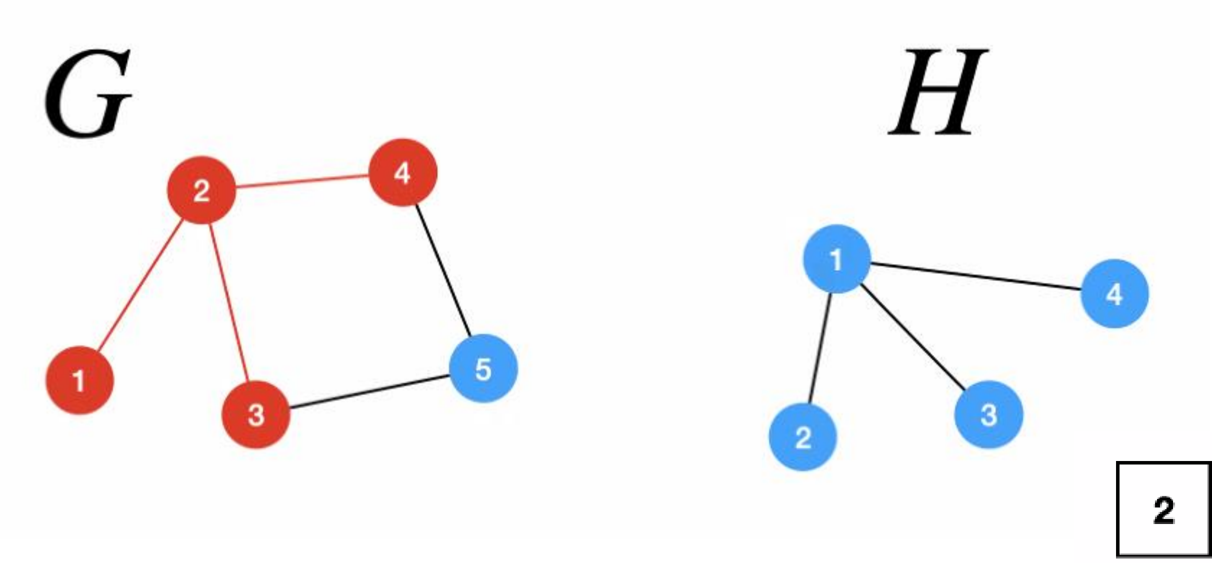
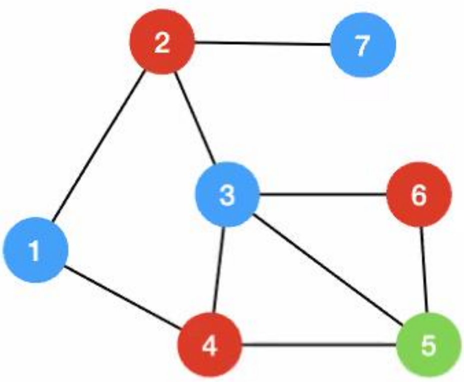
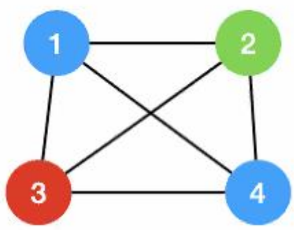


2	1	3	4
1	2	3	4

```
algo isomorphisle-sous-graphes(G = (S,A), H = (S',A')):
    n := |S|
    m := |S'|
    R := tableau de taille m
    pour i := 1 à m faire :
        R[i] := deviner (1,...,n)
    si il y a répétitions dans R alors
        ajouter
    pour chaque {u,v} ∈ A' faire
        si {R[u],R[v]} ∈ pas à S alors
            rejeter
    accepter
```

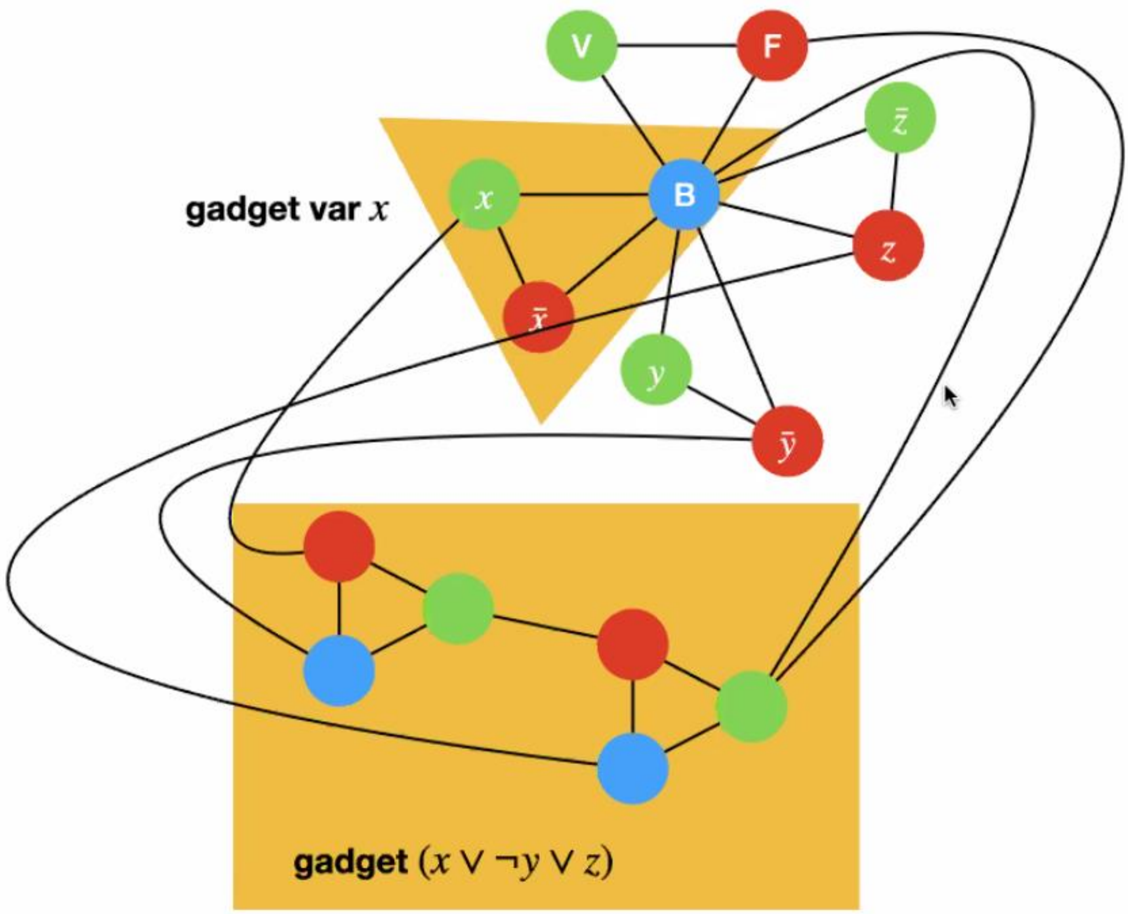




```
algo 3-colorabilité(G=(S,A)):
   n := |S|
   C := tableau de taille n
   pour i := 1 à n faire
      C[i] := deviner(rouge,bleu,vert)
   pour chaque {u,v} ∈ A faire
      si C[u] = C[v] alors
         rejeter
   accepter
```

```
algo vérificateur-3-colorabilité(G=(S,A), C):
   n := |S|
   pour chaque {u,v} ∈ A faire
      si C[u] = C[v] alors
         rejeter
   accepter
```

$$(x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee y \vee \neg z)$$



```
C = tableau avec les sommets du cycle en ordre

algo vérificateur-cycle-hamiltonien(G=(S,A), C)
   n := |S|
   si C contient des sommets dupliqués alors
      rejeter
   si C ne contient pas tous les sommets de S alors
      rejeter
   pour i := 1 à n−1 faire
      si (C[i],C[i+1]) ∉ A alors
         rejeter
   si (C[n],C[1]) ∉ A alors
      rejeter
   accepter
```