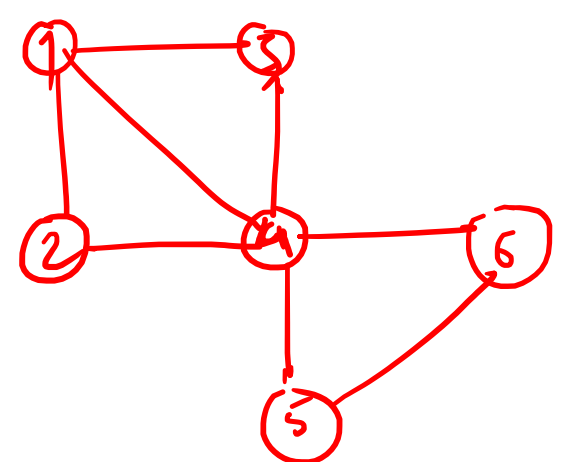
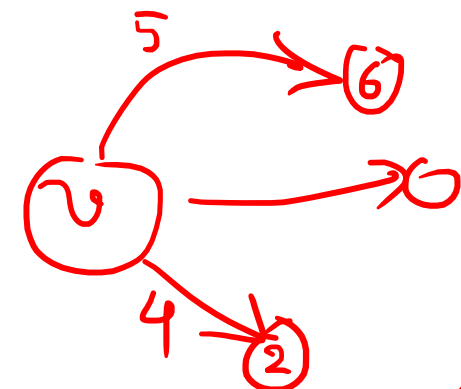


$G = (V, E)$   $V = \{1, 2, 3, 4, 5, 6\}$   
 $E = \{(1, 2), (1, 3), (1, 4), (2, 4), (3, 4), (4, 5), (4, 6), (5, 6)\}$

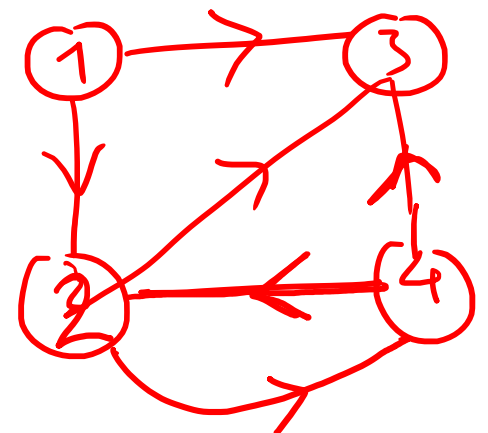


D/S hế:  $A[u]$  list các đỉnh kề với  $u$ .  
 vector<int>  $A[N]$ ;

set<int>  $\Rightarrow$  find nhanh



có hướng directed graph



$V = \{1, 2, 3, 4\}$   
 cung(arc):  $E = \{(1, 2), (1, 3), (2, 3), (2, 4), (4, 2), (4, 3)\}$

Đồ thị trên cung  
 struct Arc {  
   int nod;  
   int w;  
 };

có trọng số  $w(u, v)$ ;  
 liên  $\hookrightarrow$  1 cung kề (đi qua) khỏi 1 đỉnh (v)  
 $\Rightarrow$  vector<Arc>  $A[N]$ .

BT ứng dụng

Duyệt theo chiều rộng  $\rightarrow$  Queue

$visited[v] = True$   
nghĩa là v đã  
được thăm.

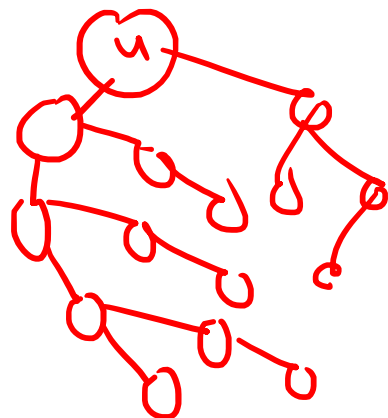
Duyệt theo chiều sâu  $\rightarrow$  Stack hoặc Đệ Quy.

DFS(u) // duyệt theo chiều sâu

Thăm u;  
 $visited[u] = True;$   $\rightarrow$  D/S kê của u

for  $v \in A[u]$  do  
if  $visited[v] = False$  then

[DFS(v);



BFS(u) // Duyệt theo chiều rộng

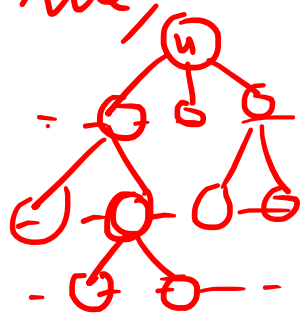
Queue Q;  
Q.push(u);  $visited[u] = True;$

while Q not empty do

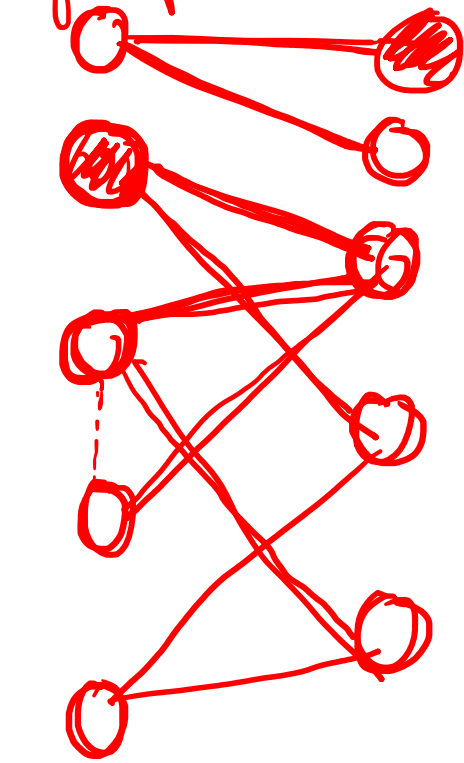
[  $v = Q.pop();$

for  $x \in A[v]$  do  
if  $visited[x] = ~~True~~ False$  then

[ Q.push(v);  
visited[v] = True;



Check if a given undirected graph is Bipartite (đồ thị 2 phía)



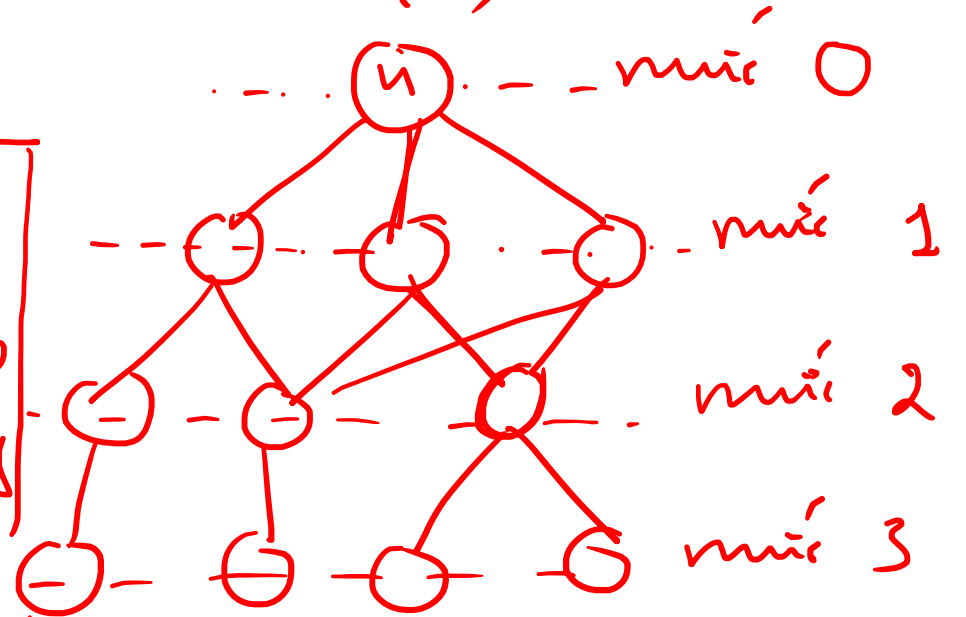
~~X~~ X



```

BFS(u)
Queue Q;
Q.push(u); d[u] = 0;
while Q not empty
    v = Q.pop();
    for x in A[v] do
        if d[x] >= 0 then
            if d[u] + d[x]
               chẵn then
                   return False
            else
                Q.push(x)
                d[x] = d[u] + 1
    return True;
    
```

BFS(u)



$d[u]$ : mức (độ dài đường ngắn nhất theo số cạnh từ đỉnh xuất phát đến u)  
 $d[u] = \begin{cases} -1 & : u \text{ chưa thăm} \\ \geq 0 & : u \text{ đã được thăm} \end{cases}$



