Esercizio 1. Un grafo diretto $G = (V, E)$ si dice diretto aciclico (DAG) se G non contiene alcun ciclo diretto. Modificare l'algoritmo della ricerca in profondità in maniera da poter controllare se un grafo diretto è aciclico o no; è possibile fare questa modifica in modo che il controllo avvenga in $\Theta(|V| + |E|)$?

Considero un array lungo |V| utile per salvare i nodi momentaneamente presenti nello stack.

```
Cerca_cicli( G:grafo, x:vertice):
    Vis[n] = {0,0..,0}
    S:Stack
    Is[n]={0,0..,0}
    Vis[x]=1
    S.push(x)
    Is[x]=1
    while(S≠∅){
        z=S.top()
        if(z.adiacenti_out==NULL){
                S.pop()
                Is[z]=0
        else{
            y= z.adiacenti_out[0]
            if(Is[y]==1){return True} // controllo un elemento gia' nello stack => ciclo
            z.adiacenti_out.remove(0)
            Vis[y]=1
            S.push[y]
            Is[y]=1
        }
    }
    return False
}
```

* Se durante la ricerca visito un nodo che e' ancora presente nello stack, c'e' un ciclo

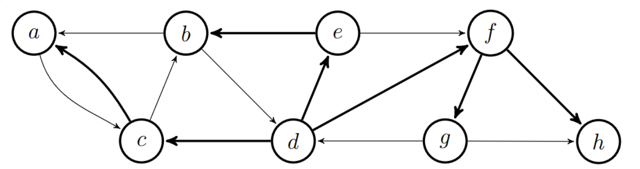
Il costo e' $O(n+m)$

Esercizio 2. Un grafo $G = (G, E)$ non diretto dice bipartito se l'insieme dei vertici V può essere partizionato in due insiemi disgiunti U e W tali che: (1) $U \cap W = \emptyset$, (2) $U \cup W = V$ e (3) ogni arco di G è incidente ad un vertice di U e ad vertice di W. E' noto che un grafo G è bipartito se e solo se G non ha cicli di lunghezza dispari. Modificare l'algoritmo della ricerca in profondità in maniera da poter controllare se un grafo non diretto è bipartito o no, e in caso fornire una bipartizione; è possibile fare questa modifica in modo che il controllo avvenga in $\Theta(|V| + |E|)$? Domanda bonus: nel caso in cui G non è bipartito, come deve essere ulteriormente modificato l'algoritmo per ritornare un ciclo dispari di G?

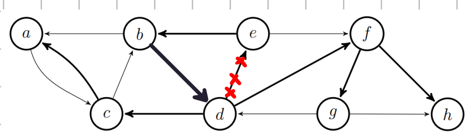
```
Bip_globale ( G:grafo )
    x = vertice di G
    COL [n] : array
    bip = True
    Vis [n] : array
    COL[x]=1
    DFS_Bip(G, COL, x, Vis, bip)
    return bip
}

DFS_Bip ( G:grafo, COL:array, x:vertice, Vis:array, bip:bool)
    Vis[x]=1
    For each(y~x){
        if(Vis[y]==0){
            COL[y]= COL[x]·-1
            DFS_Bip(G, COL, y, Vis, bip)
        }else{
            if(COL[y]==COL[x]) { bip=False }
        }
    }
}
```

Esercizio 3 (I. Salvo). Si consideri il grafo diretto G illustrato nella figura qui sotto e l'albero T formato formato dagli archi evidenziati. L'albero T può essere prodotto da una ricerca in profondità?



No, non e' Frutto di una DFS. Il seguente, si (nodo di partenza: e)



Liste adiacenza:
e: { b, f }
b: { d, a }
c: { a, b }
d: { c, f }
a: { c }
f: { h, g }
g: { h, d }