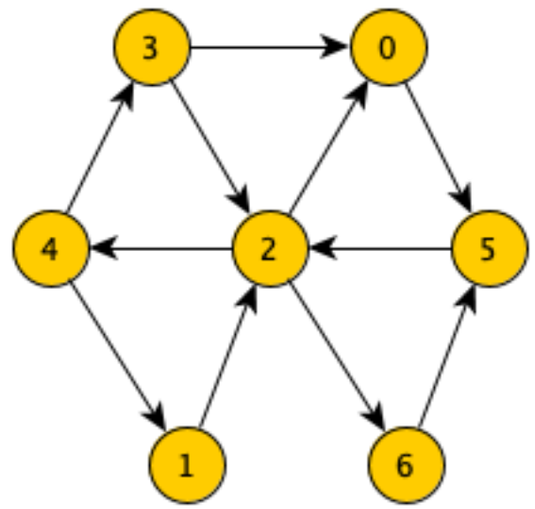
Esercizio appello straordinario aprile 2020

Si consideri il grafo diretto G rappresentato in figura:

$G =$ 

Assumiamo che G sia rappresentato tramite liste di adiacenza nelle quali i nodi compaiono in ordine crescente. Questo significa che in caso di visita del grafo nel caso un nodo abbia più nodi adiacenti da visitare viene sempre scelto quello di indice minimo.

1. Considerate una visita in profondità che parte dal nodo 2 di G.
   - Riportate nell'ordine i nodi di G che vengono effettivamente visitati.
   - Individuate gli archi in avanti, gli archi all'indietro e gli archi di attraversamento che si incontrano durante la visita.
2. Considerate una visita in ampiezza che parte dal nodo 2 di G.
   - Riportate nell'ordine i nodi di G che vengono effettivamente visitati.
3. Qual è il numero minimo di archi da eliminare da G perché il grafo risulti avere ordinamenti topologici e quali sono questi archi? **Motivare bene la risposta.**
4. Eliminate da G gli archi ottenuti al punto precedente in modo che il grafo G' ottenuto risulti avere ordinamenti topologici.
   - quanti e quali sono gli ordinamenti topologici del grafo G'? **Motivare bene la risposta.**
   - Individuate l'ordinamento topologico che si ottiene applicando a G' l'algoritmo visto a lezione basato sulle sorgenti (in caso siano disponibili più sorgenti assumete che l'algoritmo scelga sempre quella di indice minimo).
   - Individuate l'ordinamento topologico che si ottiene applicando a G' l'algoritmo visto a lezione basato sulla visita in profondità (In caso siano disponibili più nodi da visitare assumete che l'algoritmo scelga sempre quello di indice minimo).
5. Eliminate le direzioni dagli archi del grafo G in modo da ottenere un grafo G" non diretto. Determinate i ponti del grafo G". **Motivare bene la vostra risposta.**

ESERCIZIO.

Vogliamo calcolare il numero di sequenze decimali di lunghezza n in cui non appaiono cifre pari adiacenti.

Progettare un algoritmo che prende come parametro l'intero n e, in tempo $O(n)$, restituisce il numero delle sequenze cui siamo interessati.

Ad esempio:

- per $n = 1$ la risposta dell'algoritmo deve essere 10
- per $n = 2$ la risposta dell'algoritmo deve essere 75

ESERCIZIO:

Considera il seguente gioco a due:
abbiamo una pila di n monete, i due giocatori A e B si alternano nelle mosse potendo sottrarre dalla cima della pila 1, 3 o 4 monete.
Vince chi per primo elimina le ultime monete dalla pila.

Progettare un algoritmo che, dato il numero n, in $O(n)$ tempo permette di sapere se il giocatore A ha una **strategia vincente**.

Il giocatore A ha una strategia vincente se, può vincere la partita indipendentemente dalle contromosse del giocatore B.

Ad esempio:

- per $n = 2$ la risposta è $False$ infatti A è costretto a prendere una moneta e a questo punto B prende l'ultima moneta rimasta e vince.
- per $n = 5$ la risposta è $True$ infatti A può prendere 3 monete, delle due monete rimanenti B è costretto a prenderne una ed A vince eliminando l'ultima moneta rimasta.

ESERCIZIO:

Dati un intero ed m progettare un algoritmo che stampi sequenze di numeri nell'intervallo $[1,\ldots,m]$ dove ogni numero è almeno il doppio del precedente. Le sequenze da stampare devono essere massimali vale a dire sequenze che non è possibile allungare ulteriormente.

Ad esempio:

- per ed $m = 6$ l'algoritmo deve stampare le seguenti sequenze: [1,2,4], [1,2,5], [1,2,6], [1,3,6], [1,4], [1,5], [1,6], [2,4], [2,5], [2,6], [3,6], [4], [5], [6].

L'algoritmo proposto deve avere complessità $O\left(S(m)\cdot \log m\right)$ dove $S(m)$ è il numero di sequenze da stampare.
Motivare **bene** la correttezza e la complessità dell'algoritmo proposto.