Corso di Progettazione di algoritmi

Esercizi 3

Esercizio 1

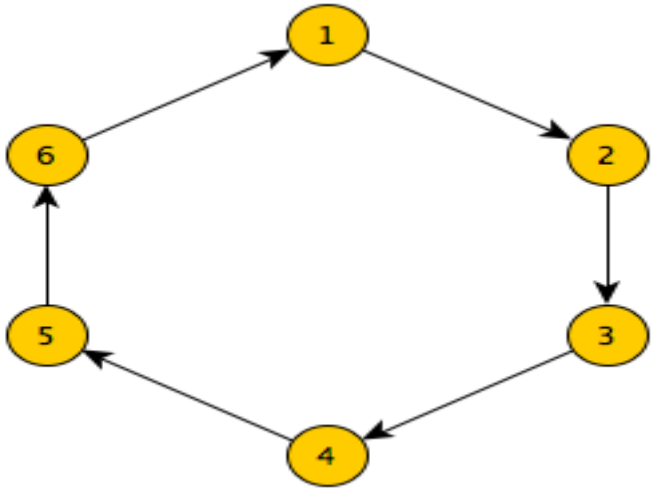
- Il seguente algoritmo, dato un grafo diretto G con nodi numerati da 1 a n ed un suo nodo u, dovrebbe ritornare $True$ se la parte di G raggiungibile da u è aciclica:

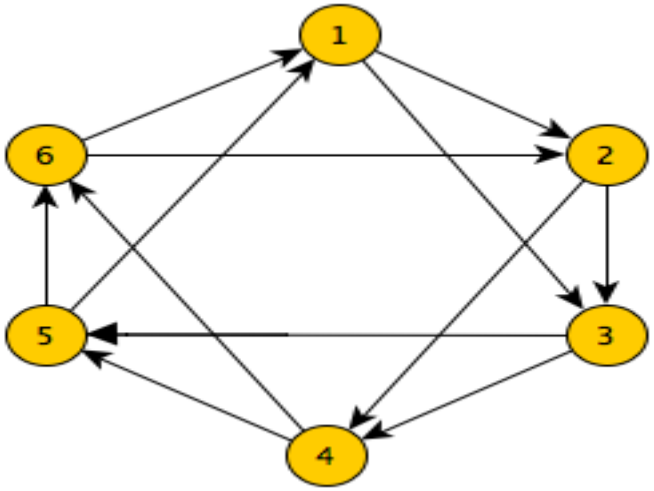
```
ACI(G, u)
    P : vettore di lunghezza n, inizializzato a −1
    Q : coda inizializzata con il solo nodo u
    P[u] ← 0
    WHILE Q non è vuota DO
        v ← estrae il primo nodo della coda Q
        FOR ogni discente w di v DO
            IF (P[w] ≠ −1) THEN
                    z ← v
                    WHILE z ≠ u AND z ≠ w DO
                            z ← P[z]
                    IF (z = w) THEN RETURN False
            ELSE
                    P[w] ← v
                    inserisci w in coda a Q
    RETURN True
```

- Dire se l'algoritmo è corretto. Più precisamente, se è corretto spiegare perché lo è, se invece non è corretto fornire un controesempio.

Esercizio 2

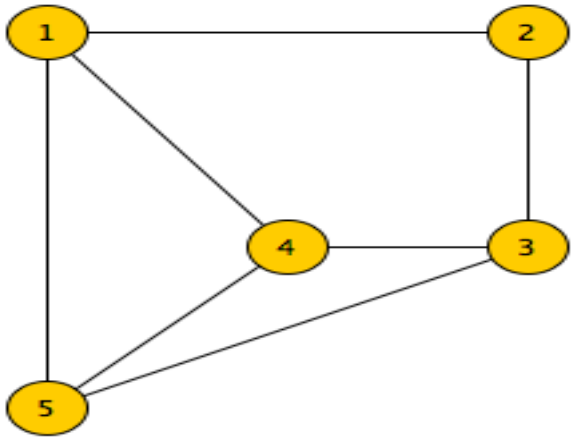
Il *grafo quadrato* di un grafo diretto $G = (V, E)$ è un grafo diretto $G^2 = (V, E^2)$ che ha lo stesso insieme di vertici e un arco da u a v se e solo se in G il vertice v è raggiungibile da u con un cammino di lunghezza due, vale a dire $E^2 = \{(u, v) | \exists w \in V, (u, w)\ e\ (w, v) \in E\}$.
Descrivere un algoritmo che, dato il grafo diretto G, ne calcola il grafo quadrato G^2 e calcolarne la complessità. Cosa cambia se il grafo è rappresentato tramite matrice di adiacenza o liste di adiacenza?





Esercizio 3

Descrivere un algoritmo che, dato un grafo connesso G, trova un cammino in G che attraversa tutti gli archi una e una sola volta in ognuna delle due direzioni. L'algoritmo deve avere complessità $O(m)$.

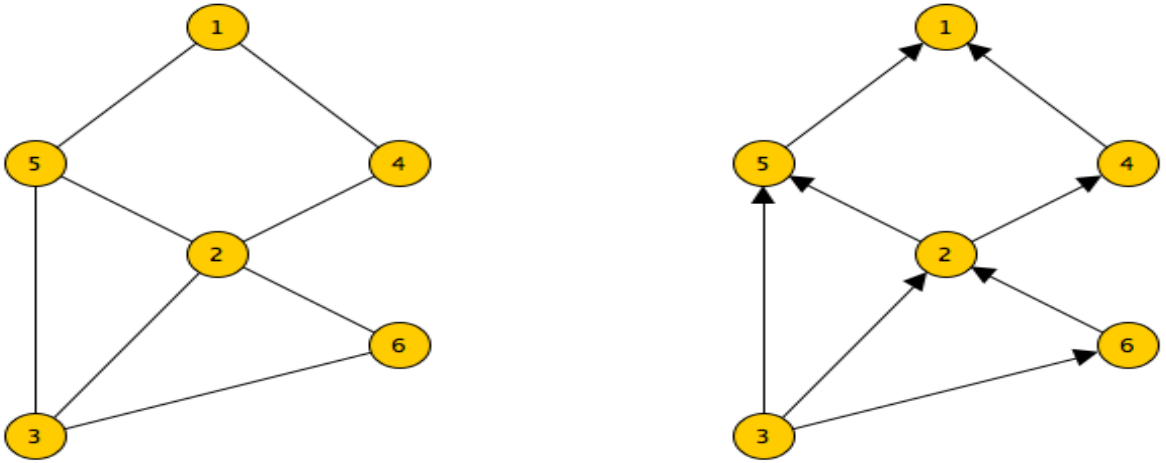


Ad esempio per il grafo in figura che ha 7 archi una possibile soluzione è il seguente cammino di lunghezza 14:

$1 - 4 - 5 - 4 - 1 - 5 - 1 - 2 - 3 - 4 - 3 - 5 - 3 - 2 - 1$

Esercizio 4

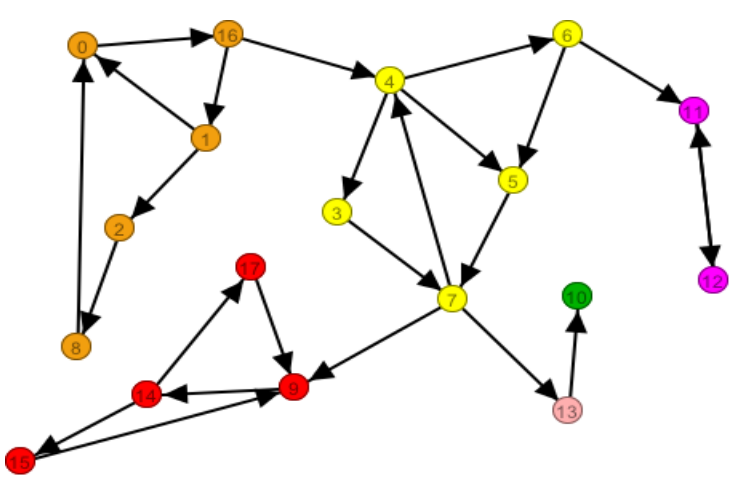
Dato un grafo G, descrivere un algoritmo che ne orienta gli archi in modo da creare un grafo G' diretto e aciclico. L'algoritmo deve avere complessità $O(n+m)$.

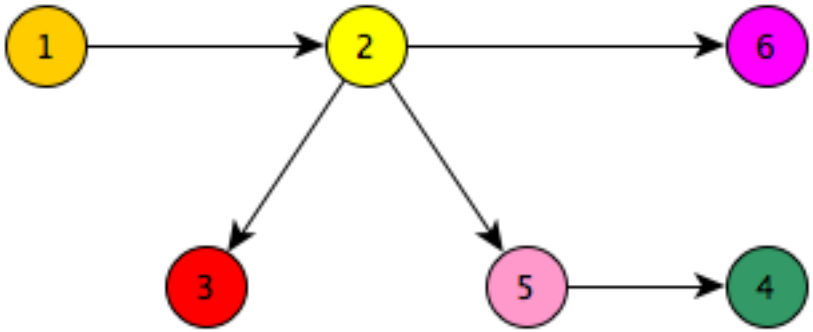


Ad esempio per il grafo sopra a sinistra un orientamento degli archi lecito è quello riportato sopra a destra

Esercizio 5

Dato un grafo diretto G, si definisce grafo delle parti il grafo G' che contiene un vertice per ogni componente fortemente connessa di G e tra due suoi nodi a e b c'è un arco che va da a a b se in G è possibile andare da un nodo della componente fortemente connessa corrispondente ad a ad un nodo della componente fortemente connessa corrispondente a b.





- Descrivere un algoritmo che, a partire dal grafo diretto G, costruisce il suo grafo delle parti G' in $O(n+m)$ tempo.

- Dimostrare che il grafo delle parti è sempre un DAG (vale a dire un grafo diretto aciclico).

Esercizio 6

Un vertice v in un grafo diretto G, si dice *principale* se ogni altro vertice in G può essere raggiunto con un cammino diretto che parte da v.

a) Descrivere un algoritmo che dati un grafo G e un vertice v, determina se v è un vertice principale in G. L'algoritmo deve avere complessità $O(n+m)$.

b) Descrivere un algoritmo che, dato un grafo G, determina se G contiene un vertice principale. L'algoritmo deve avere complessità $O(n+m)$.

Esercizio 7

Descrivere un algoritmo che dato un grafo diretto G trova il minimo numero di vertici da cui è possibile raggiungere tutti gli altri vertici del grafo. L'algoritmo deve avere complessità $O(n + m)$.

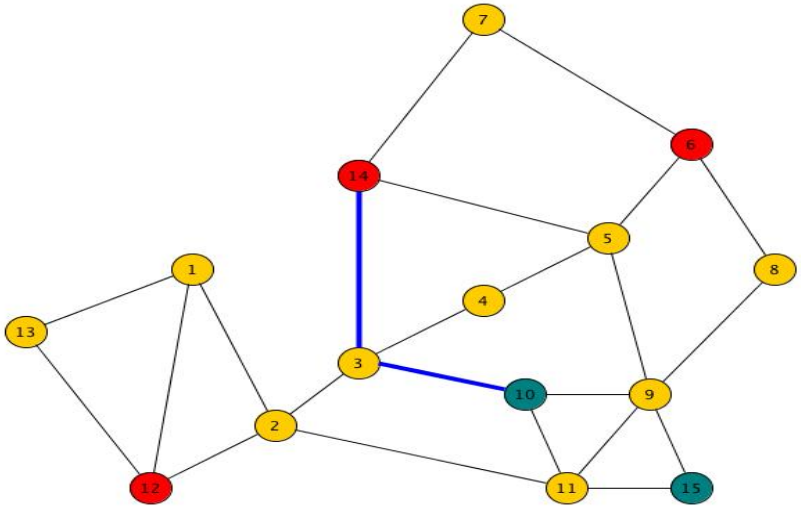
Esercizio 8

Descrivere un algoritmo che, dato un grafo G non diretto e connesso e due suoi nodi u e v, in tempo $O(n+m)$ trova i nodi che hanno la stessa distanza da u e v.

Esercizio 9

Dato un grafo G e due sottoinsiemi V_1 e V_2 dei suoi vertici si definisce distanza tra V_1 e V_2 la distanza minima per andare da un nodo in V_1 ad un nodo in V_2. Nel caso V_1 e V_2 non sono disgiunti allora il valore 0.
Descrivere un algoritmo che, dato un grafo G e i due sottoinsiemi dei vertici V_1 e V_2 calcola la loro distanza. L'algoritmo deve avere complessità $O(n+m)$.

- Ad esempio per il grafo G in figura, dove i nodi dell'insieme A sono in verde mentre i nodi dell'insieme B sono in rosso, la distanza tra i due insiemi è 2 come evidenziato dal cammino in blu.



Esercizio 10

Dare lo pseudo-codice di un algoritmo che preso in input un grafo non diretto e connesso G, un suo nodo u, un vettore dei padri P relativo a una BFS da u in G e un arco $\{v, w\}$ di G, ritorna $True$ se e solo se la rimozione dell'arco $\{v, w\}$ non cambia le distanze da u. L'algoritmo deve avere complessità $O(n)$.