L’algoritmo hasCycleDFS.

Il file HCDFS\_module.py contiene l’implementazione dell’algoritmo hasCycleDFS (HCDFS)e le relative funzioni che consentono il suo funzionamento. Partendo proprio dalla funzione HCDFS(), essa assume in input il parametro G, ovvero il grafo, ed altri dati opzionali quali:

1. Debug, booleano che permette di visualizzare informazioni in più per il debugging;
2. showProfile, per la visualizzazione a schermo delle informazioni che vengono scritte dal Decorator all’interno del file (Vedi sezione D\_module).

Al suo interno viene creata un’istanza di CustomGAM (CGAM), ovvero una sottoclasse della classe GraphAdjacencyMatrix. Si è scelto di implementare questa sottoclasse per non alterare la struttura del metodo dfs() presente in Graph.py per la rilevazione di cicli nei grafi.

In seguito viene chiamato il metodo dfsDetectedCycle() presente in CustomGAM, che prende in input i parametri rootId (nodo di partenza della visita DFS) e debug. Esso controllerà attraverso la visita DFS se è presente o meno un ciclo in G.

L’idea alla base della realizzazione dell’algoritmo è stata la seguente:

La visita DFS viene implementata attraverso uno stack e vengono aggiunti al suo interno tutti i nodi adiacenti al vertice preso in considerazione che non sono ancora stati esplorati. Questo vuol dire che se un nodo (A) è presente all’interno dello stack almeno 2 volte, allora esso è stato “visto” da due nodi diversi ma a lui adiacenti (B, C adiacenti ad A). Perché questo dovrebbe implicare la presenza di un ciclo? Perché B e C sono connessi tra loro o attraverso un arco oppure attraverso un altro nodo. Infatti se ci siamo fermati sul nodo B, abbiamo “visto” tutti i suoi nodi adiacenti, abbiamo messo A nello stack ed in seguito siamo passati al nodo C, allora esso deve necessariamente essere connesso a B, direttamente o attraverso un altro vertice, in quanto il nodo successivo da esaminare è scelto attraverso un pop dalla pila stessa.

Quindi viene effettuato un controllo sullo stack ed attraverso un contatore si esamina quante volte è presente un dato nodo al suo interno.