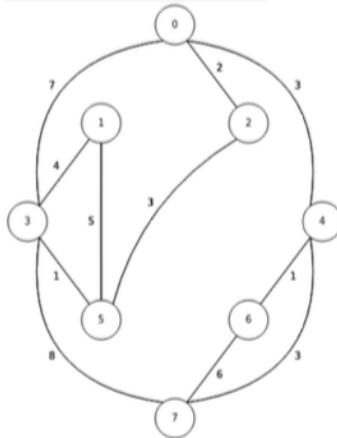


Soluzione Grafi

Visita DFS

Si consideri il seguente grafo pesato e non orientato:



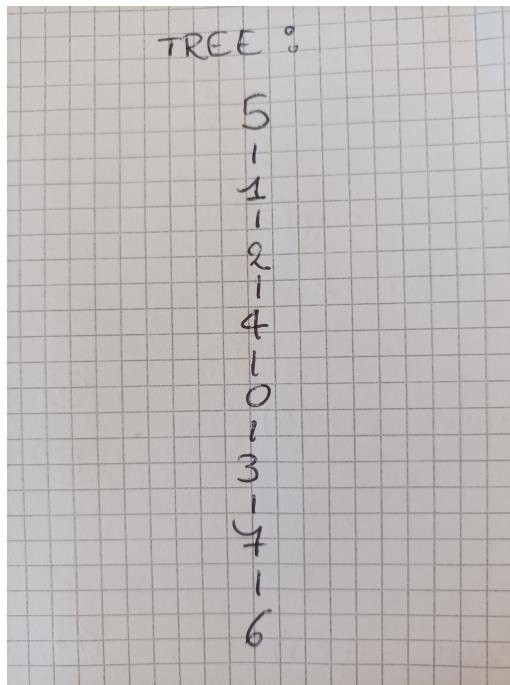
[2/3 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita DFS (non importa se implementata mediante algoritmo ricorsivo o iterativo: la visita deve sempre andare in profondità) di tale grafo che parta dal nodo etichettato con **5** con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.

Parto da **nodo 5**:

Stack = 5 Tree_Root = 5

Visito	Stack	Parent
adj(5)	1 - 5	P[1] = 5
adj(1)	2 - 1 - 5	P[2] = 1
adj(2)	4 - 2 - 1 - 5	P[4] = 2
adj(4)	0 - 4 - 2 - 1 - 5	P[0] = 4
adj(0)	3 - 0 - 4 - 2 - 1 - 5	P[3] = 0
adj(3)	7 - 3 - 0 - 4 - 2 - 1 - 5	P[7] = 3
adj(7)	6 - 7 - 3 - 0 - 4 - 2 - 1 - 5	P[6] = 7
adj(6) --> non ci sono altri elementi	(backtracking) 7 - 3 - 0 - 4 - 2 - 1 - 5	/

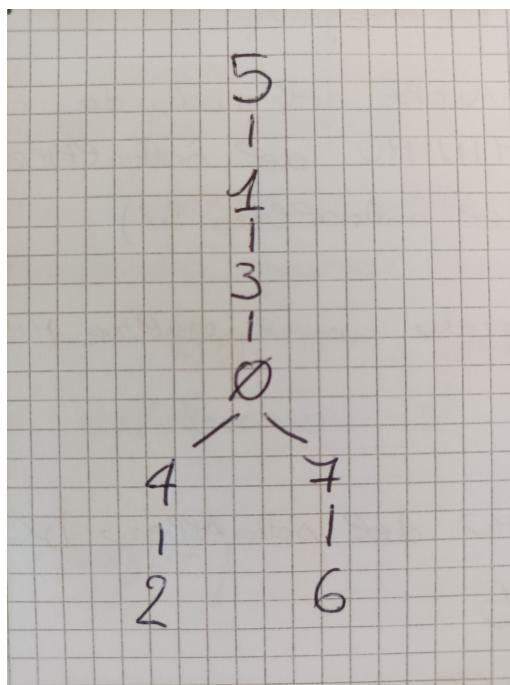
Da questo momento in poi, non essendoci altri nodi ancora da visitare, l' algoritmo esegue il backtracking delle chiamate ricorsive e una volta terminato lo stack, ci restituisce il seguente albero di visita DFS:



[1/3 del punteggio] Se è possibile, si disegni un albero di ricoprimento differente da quello ottenuto prima, sempre generato da una visita DFS del grafo che parta dal nodo **5** (senza illustrare i passaggi con cui viene generato). Se non è possibile ottenere un albero di ricoprimento diverso, se ne spieghi la ragione

A seconda dell'ordine con cui vengono scelti ed estratti gli archi, è possibile ottenere più di un unico albero di visita DFS.

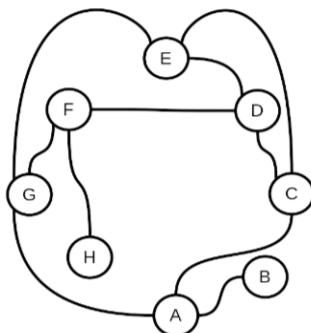
Nell'esempio sotto, quando si esplorano i nodi adiacenti al nodo **1**, invece di scegliere il nodo **2** come nell'esercizio sopra, si opta per il nodo **3**. Questo ci porta ad avere il seguente albero:



Visita BFS

!! Ogni vertice è marcato al più una volta, aggiunto dalla coda una volta, cancellato dalla coda una volta !!

[1/2 del punteggio] Si illustrino mediante disegni o altra modalità schematica e chiara i vari passaggi di una visita **BFS** del seguente grafo che parta dal nodo etichettato con **C** con creazione dell'albero di ricoprimento; si disegni l'albero di ricoprimento risultante.



Queue = C Tree_root = C

Visito	Queue (old --> new)	Parent
adjList(C)	C --> A - D - E	P[A,D,E] = C
adjList(A)	A - D - E --> D - E - B - G	P[B,G] = A
adjList(D)	D - E - B - G --> E - B - G - F	P[F] = D
adjList(E)	E - B - G - F --> B - G - F	
adjList(B)	B - G - F --> G - F	
AdjList(G)	G - F --> F	
AdjList(F)	F --> H	P[H] = F
AdjList(H)	H --> /	

Albero risultante:

