

Esercitazione 10: Final Problem Session

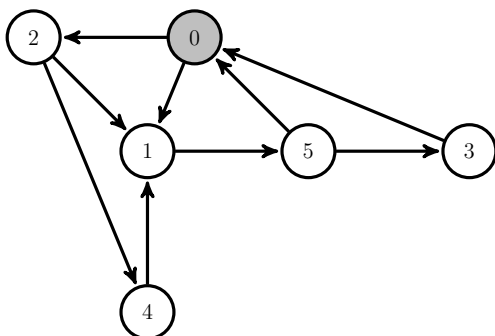
Giacomo Paesani

May 25, 2026

Esercizio 1. Dato un albero T di n nodi, rappresentato tramite il vettore dei padri P , dare lo pseudo-codice di un algoritmo che produce in tempo $\mathcal{O}(n)$ la lista dei vertici di T_u , il sotto-albero radicato in un vertice u di T .

Esercizio 2. Sia G il grafo in figura in cui le liste di adiacenza sono ordinate in senso crescente degli indici. Allora, determinare:

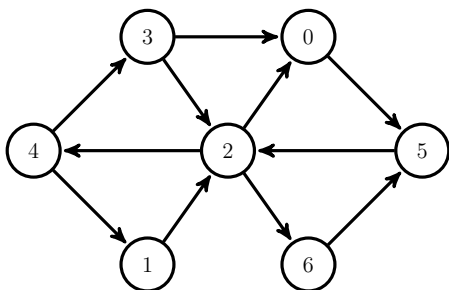
- l'albero di ricerca ottenuto in seguito ad una ricerca in profondità (DFS) di G con radice il vertice 0;
- specificare quali sono gli archi di attraversamento, in avanti e all'indietro in seguito a tale DFS.



Esercizio 3. Sia G il grafo in figura in cui le liste di adiacenza sono ordinate in senso crescente degli indici.

1. Considerare una visita in profondità (DFS) con radice il vertice 2, allora:

- 1a. riportare nell'ordine i vertici di G che vengono effettivamente visitati;
- 1b. individuare gli archi in avanti, all'indietro, di attraversamento che sono individuati durante la visita.
2. Considerare una visita in ampiezza (BFS) con radice il vertice 2, allora riportare nell'ordine i vertici di G che vengono effettivamente visitati.
3. Qual'è il minimo numero di archi da eliminare da G perché il grafo ottenuto risulti avere ordinamenti topologici e quali sono questi archi?
4. Eliminare da G gli archi ottenuti al punto precedente in modo che il grafo ottenuto G' risulti avere ordinamenti topologici, allora determinare quanti e quali sono gli ordinamenti topologici di G' .
5. Eliminare le direzioni degli archi da G ottenendo un grafo G'' non diretto. Determinare i ponti di G'' .



Esercizio 4. Dato un intero $n \geq 2$, definiamo con x_n il minimo numero di operazioni con cui è possibile ottenere n partendo dal numero 2 e potendo effettuare le sole 3 operazioni: di incremento di 1, prodotto per due e prodotto per tre; e con y_n il numero totale di modi (non per forza con un numero minimo di operazioni) per ottenere n con tali operazioni. Fornire un algoritmo che dato un intero n calcola sia x_n che y_n in tempo $\mathcal{O}(n)$.