8 Grafi non orientati, BFS, DFS

8.1 Dimostrazioni non induttive

Dimostrare le seguenti proprietà:

1. In un grafo con almeno due nodi, ci sono sempre due nodi con lo stesso grado.

2. Dato un grafo $G = (V, E)$, definiamo il *grafo complementare* il grafo $\overline{G} = (\overline{V}, \overline{E}$, dove $\overline{V} = V$ e $\overline{E} = \{(u, v) \mid (u, v) \notin E\}$. Dimostrare che almeno uno tra G e $\overline{G}$ è connesso.

8.2 Capire la BFS e la DFS

Rispondere ai seguenti quesiti:

1. Fare un esempio di un grafo $G = (V, E)$, una sorgente $s \in V$ e un insieme di archi $E' \subseteq E$ tali che $T = (V, E')$ sia un albero ricoprente di G e per ogni vertice in $v \in V$ l'unico cammino semplice da s a v sia minimo in G, ma T non sia il risultato di nessuna BFS radicata in s (indipendentemente dall'ordinamento delle liste di adiacenza).

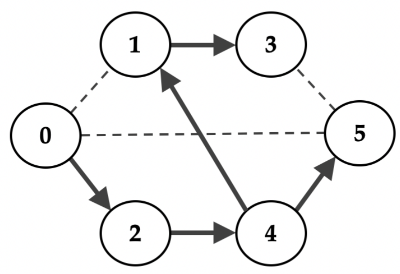
2. Come deve essere fatto un grafo in modo che l'albero di visita della BFS sia esattamente uguale all'albero di visita di una DFS?

8.3 BFS vs DFS (Esame 17/2/23)

Dato il grafo in figura, dire perché gli archi disegnati con frecce continue in grassetto non possono essere né l'albero di visita di una visita in profondità (DFS) né quello di una visita in ampiezza (BFS) radicata in 0.

Modificate il meno possibile l'albero in modo che sia il possibile risultato di una DFS indicando un possibile ordine in cui sono percorse le liste di adiacenza per ottenere tale albero di visita.

Dare infine la visita BFS corrispondente a quelle stesse liste di adiacenza.



8.4 Rappresentazioni

Considerate le 4 possibili rappresentazioni in memoria di un grafo. Dare degli algoritmi per tradurre un grafo da una rappresentazione all'altra.

Stabilite degli (evidenti) limiti inferiori per ciascuna di queste traduzioni (determinate dalla necessit di leggere l'input oppure di scrivere l'output) e verificate se il vostro algoritmo sia ottimo o meno.

8.5 Nodi a distanza massima (esame 1/9/22)

Dato un grafo non orientato e connesso $G = (V, E)$ e un nodo $s \in V$, considerare il problema di determinare i nodi a distanza massima da s in V, dove la distanza di un nodo u da s è il numero di archi di un cammino minimo da s a u.

Scrivere lo pseudo-codice di una funzione `u, dmax, k = distMax(G, s)` che carica in `u` un qualsiasi nodo a distanza massima, in `dmax` il valore della distanza massima, e in `k` il numero di nodi a distanza massima.

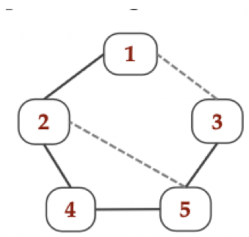
FACOLTATIVO: invece di un nodo u, tornare la lista di tutti i nodi a distanza massima.

8.6 Distanze minime: archi neri e archi rossi (IV esonero, 16/1/23)

In un grafo non orientato $G = (V, E_R \cup E_N)$, gli archi sono sono partizionati in archi rossi (E_R) e neri (E_N). Sapendo che percorrere un arco rosso costa 0 e percorrere un arco nero costa 1, fissato un nodo s, si vuole determinare la distanza (cioè il minimo numero di archi neri che è necessario percorrere) di tutti i nodi da s. Dovete:

1. Calcolare le distanze dal nodo 1 nel grafo in Figura, dove gli archi rossi sono tratteggiati;

2. dare un algoritmo per determinare le distanze da un certo nodo s (assumete di avere due liste di adiacenza per ciascun nodo v, $adjR[v]$ e $adjN[v]$).



8.7 Distanze minime tra due insiemi di nodi

Dati due insiemi di nodi U, Z in un grafo non orientato $G = (V, E)$ $(U, Z \subseteq V)$, definiamo la distanza $d(U, Z)$ come $min_{u \in U, z \in Z} d(u, z)$.

Proporre un modo per rappresentare i due insiemi U e V.

Descrivere un algoritmo per calcolare $d(U, Z)$ (è possibile dare una soluzione $\mathcal{O}(m + n)$).