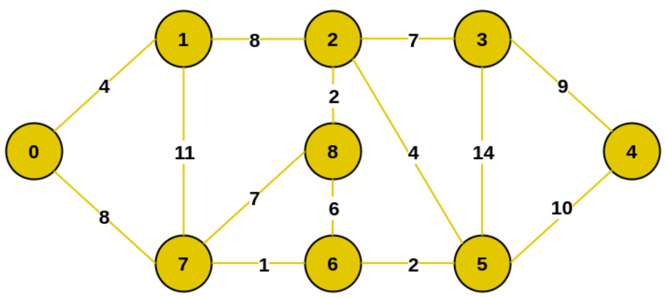
Analisi e progettazione di algoritmi

(III anno Laurea Triennale - a.a. 2022/23)

Prova scritta 18 gennaio 2024

Esercizio 1 Si esegua, sul seguente grafo:



l'algoritmo di Prim a partire dal nodo 0. Inizialmente quindi si avrà `dist`$(0)=0$ e `dist`$(u)=\infty$ per tutti gli altri nodi. Per ogni iterazione del ciclo while si dia:

- il nodo che viene estratto con la `getMin`
- i nodi per i quali viene modificata `dist` e come
- l'albero ottenuto alla fine dell'iterazione, evidenziando chiaramente la parte di albero definitiva.

In caso di più scelte possibili si segua l'ordine numerico.

Soluzione Rappresento l'albero come lista di coppie per mia comodità (scrivendo a mano è più comodo un disegno).

getMin	0	1	2	3	4	5	6	7	8	*albero*
	0	∞	∞	∞	∞	∞	∞	∞	∞	
0		4						8		$(0,1), (0,7)$
1			8							**(0,1)**, $(0,7), (1,2)$
2				7		4			2	**(0,1)**, $(0,7)$, **(1,2)**, $(2,3), (2,5), (2,8)$
8							6	7		**(0,1)**, **(1,2)**, $(2,3), (2,5)$, **(2,8)**, $(8,6), (8,7)$
5					10		2			**(0,1)**, **(1,2)**, $(2,3)$, **(2,5)**, **(2,8)**, $(5,4), (5,6), (8,7)$
6								1		**(0,1)**, **(1,2)**, $(2,3)$, **(2,5)**, **(2,8)**, $(5,4)$, **(5,6)**, $(6,7)$
7										**(0,1)**, **(1,2)**, $(2,3)$, **(2,5)**, **(2,8)**, $(5,4)$, **(5,6)**, **(6,7)**
3					9					**(0,1)**, **(1,2)**, **(2,3)**, **(2,5)**, **(2,8)**, $(3,4)$, **(5,6)**, **(6,7)**
4										**(0,1)**, **(1,2)**, **(2,3)**, **(2,5)**, **(2,8)**, **(3,4)**, **(5,6)**, **(6,7)**

Esercizio 2 Rispondere alle seguenti domande, giustificando le risposte.

1. Consideriamo un grafo orientato connesso con nodi A, B, C, D, E.
   (a) Si diano, se possibile, archi e pesi in modo che l'algoritmo di Dijkstra non effettui mai il confronto tra distanze (dist[u] + $\mathtt{c_{u,v}}$ < dist[v])
   (b) Si diano, se possibile, archi e pesi in modo che nell'algoritmo di Dijkstra il nodo E cambi distanza provvisoria 4 volte.

2. Supponiamo ora che nel grafo esistano i seguenti archi: $A \to B \to C \to D \to E$
   (a) si aggiungano archi in modo che il grafo abbia due componenti fortemente connesse
   (b) si tolgano archi in modo che il grafo abbia esattamente due ordini topologici

3. Consideriamo il problema SAT, e, data una formula ϕ, sia n il numero di variabili che compaiono in ϕ, m la dimensione (ossia, lunghezza) di ϕ.
   (a) si dia un semplice limite inferiore alla complessità del problema SAT
   (b) si dia un semplice limite superiore alla complessità del problema SAT

Soluzione

1. (a) Non è possibile in quanto essendo il grafo connesso si confronterà, per ogni nodo escluso quello sorgente, la distanza iniziale ∞ con un peso. Se si esclude il primo confronto con ∞ va bene qualunque grafo in cui ogni nodo (escluso quello sorgente) abbia un solo arco entrante (ho dato metà punti per questa soluzione).
   (b) Per esempio, $A \to^0 B \to^0 C \to^0 D \to^0 E$, $A \to^3 E$, $B \to^2 E$, $C \to^1 E$, $D \to^1 E$.

2. (a) Per esempio, $C \to A$ e $E \to D$.
   (b) Non è possibile in quanto si vede facilmente che togliendo un qualunque arco si ottengono più di due ordini topologici.

3. (a) Un semplice limite inferiore è dato dal fatto che si deve necessariamente esaminare tutta la formula, quindi $\Omega(m)$.
   (b) Un semplice limite superiore è dato dall'algoritmo brute-force che prova tutte le assegnazioni di valori di verità alle variabili e per ognuna di queste valuta la formula, quindi $O(2^n \cdot m)$.

Esercizio 3

1. Date le matrici
$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 \\ -2 & 1 \end{pmatrix}$$
verifica se $AB = C$ usando *MCMatrixMultiplicationVerifier*.

2. Se A, B e C fossero 1000×1000, determina quante verifiche sarebbero necessarie per affermare che $AB = C$ con probabilità pari al 99.9%.