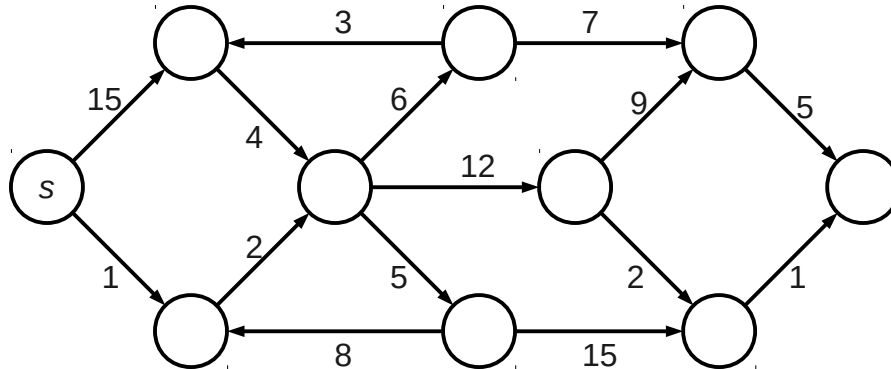


Corso di Algoritmi e Strutture Dati—Modulo 2

Esercizi su grafi (cammini di costo minimo) – 8/4/2025

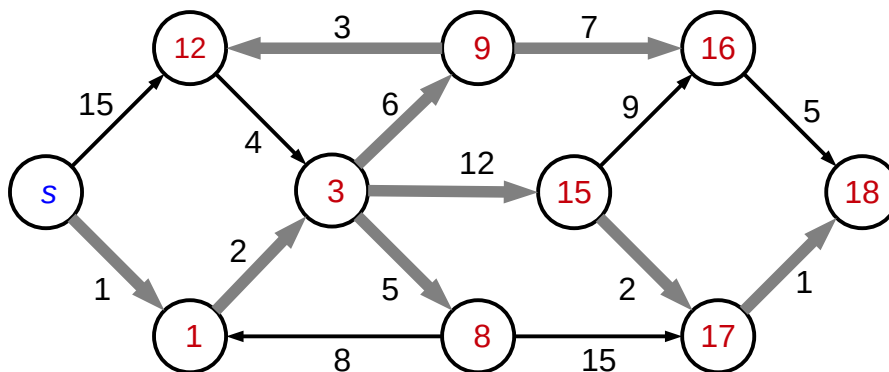
Moreno Marzolla, Jocelyne Elias

Esercizio 1. Si consideri il grafo orientato $G = (V, E)$, ai cui archi sono associati *costi positivi* come illustrato in figura:



Applicare “manualmente” l'algoritmo di Dijkstra per calcolare un albero dei cammini di costo minimo partendo dal nodo sorgente s . Mostrare il risultato finale dell'algoritmo di Dijkstra, inserendo all'interno di ogni nodo la distanza minima da s , ed evidenziando opportunamente (ad esempio, cerchiando il valore del costo) gli archi che fanno parte dell'albero dei cammini di costo minimo.

Soluzione.



Esercizio 2. Una rete stradale è descritta da un grafo orientato pesato $G = (V, E, w)$. Per ogni arco (u, v) , la funzione costo $w(u, v)$ indica la quantità di carburante (in litri) che è necessario consumare per percorrere la strada che va da u a v . Tutti i costi sono strettamente positivi. Un veicolo ha il serbatoio in grado di contenere C litri di carburante, inizialmente completamente pieno. Non sono presenti distributori di carburante. Scrivere un algoritmo efficiente che, dati in input il grafo pesato G , la quantità C di carburante inizialmente presente nel serbatoio, e due nodi s e d , restituisce *true* se e solo se esiste un cammino che consente al veicolo di raggiungere d partendo da s , senza esaurire il carburante durante il tragitto.

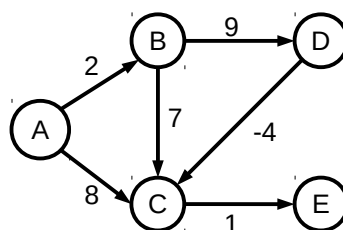
Soluzione. È sufficiente eseguire l'algoritmo di Dijkstra e calcolare il vettore $D[v]$ delle distanze minime dalla sorgente s a ogni nodo v raggiungibile da s , utilizzando il consumo di carburante come peso. È possibile interrompere l'esecuzione dell'algoritmo di Dijkstra appena si raggiunge d oppure si supera il costo C . L'algoritmo restituisce *true* se e solo se si raggiunge d e non si è superato C .

Esercizio 3. Si consideri un grafo orientato pesato, composto dai nodi $\{A, B, C, D, E\}$, la cui matrice di adiacenza è la seguente (le caselle vuote indicano l'assenza dell'arco corrispondente; l'intestazione di ogni riga indica il nodo sorgente, mentre l'intestazione della colonna indica il nodo destinazione):

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>		2	8		
<i>B</i>			7	9	
<i>C</i>					1
<i>D</i>			-4		
<i>E</i>					

1. Disegnare il grafo corrispondente alla matrice di adiacenza.
2. Determinare la distanza minima di ciascun nodo dal nodo sorgente A . Quale degli algoritmi visti a lezione può essere impiegato?

Soluzione. Il grafo è il seguente



Si nota come l'arco (D, C) abbia peso negativo; quindi per calcolare le distanze minime non è possibile usare l'algoritmo di Dijkstra, e si deve ricorrere ad esempio all'algoritmo di Bellman-Ford. Le iterazioni dell'algoritmo sono indicate nel seguito (i nodi in grigio sono quelli la cui distanza cambia):

