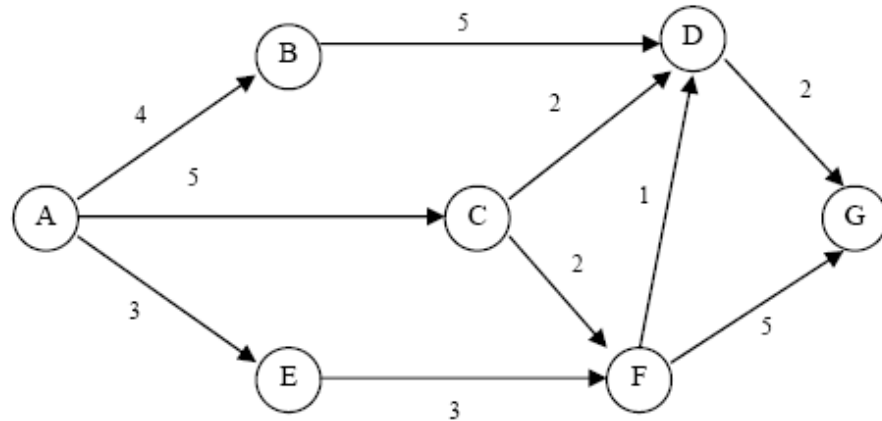


Svolgere esattamente 3 dei seguenti esercizi:

1)

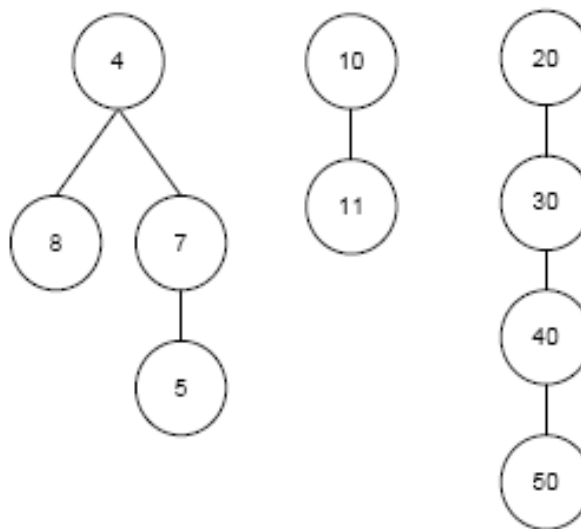
Si consideri il seguente grafo orientato e pesato



Trovare le distanze dal nodo A ad ogni altro nodo simulando l'esecuzione dell'algoritmo di Bellman-Ford.

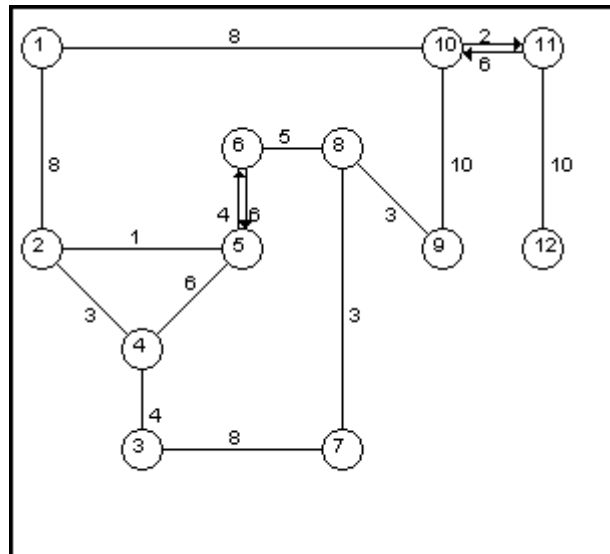
2) Si considerino le 2 sequenze $X = \langle A, A, D, C \rangle$ e $Y = \langle A, D, B, C \rangle$. Calcolare la sottosequenza più lunga comune alle 2 stringhe X e Y, simulando l'esecuzione dell'algoritmo noto basato sulla programmazione dinamica.

3) Si consideri la seguente foresta di alberi Quick Union:



Seguendo le regole dettate dall'euristiche di bilanciamento union by rank e path compression, rappresentare tutte le modifiche apportate alla foresta in seguito all'applicazione della seguente sequenza di operazioni: $union(10, 4)$, $find(50)$, $union(20, 10)$.

4) Si consideri il seguente grafo orientato e pesato:



in cui un arco non orientato di peso x sta ad indicare due archi orientati aventi lo stesso peso x .
Si supponga che sia stato preventivamente calcolato il vettore $d[]$ delle distanze dal nodo 1 ad ogni altro nodo, ottenendo:

$d[1]=0, d[2]=8, d[3]=15, d[4]=11,$
 $d[5]=9, d[6]=13, d[7]=21, d[8]=18,$
 $d[9]=18, d[10]=8, d[11]=10, d[12]=20$

Calcolare il cammino minimo dal nodo 1 al nodo 7.

Rispondere ad esattamente 3 delle seguenti domande:

1. Nel contesto del problema della Ricerca dei Cammini Minimi, illustrare e discutere l'algoritmo per costruire i cammini minimi, qualora siano note le distanze dalla sorgente x ad ogni altro vertice.
2. Illustrare e discutere l'algoritmo (basato sulla programmazione dinamica) utilizzato per ottenere la disposizione delle parentesi che minimizza il numero di moltiplicazioni elementari nel prodotto di n matrici assegnate.
3. Nel contesto del problema della Ricerca dei Cammini Minimi, illustrare e discutere l'algoritmo di Bellman e Ford.
4. Dimostrare il seguente Teorema: Sia $G=(V, E)$ un grafo non orientato, connesso e pesato. Siano, inoltre, m ed n rispettivamente il numero di archi ed il numero dei vertici del grafo. La complessità dell'algoritmo di Kruskal nel caso peggiore è $O(m \log n)$.