

Esercizio 1

Sia data la sequenza di interi, supposta memorizzata in un vettore:

32 45 55 61 21 29 39 38 37 33

1. La si trasformi in un heap, ipotizzando di usare un vettore come struttura dati. Si riportino graficamente i diversi passi della costruzione dell'heap ed il risultato finale. Si ipotizzi che, alla fine, nella radice dell'heap sia memorizzato il valore minimo. Si eseguano su tale heap i primi 2 passi dell'algoritmo di heapsort. **NB:** la sequenza è già memorizzata nel vettore e rappresenta una configurazione intermedia per cui la proprietà di heap non è ancora soddisfatta.
2. Si eseguano (sulla stessa sequenza iniziale) i primi 2 passi dell'algoritmo di quicksort per ottenere un ordinamento ascendente. **NB:** i passi sono da intendersi, impropriamente, come in ampiezza sull'albero della recursione, non in profondità. Si chiede, pertanto, che siano ritornate le 2 partizioni del vettore originale e le due partizioni di ciascuna delle partizioni trovate al passo precedente.

SOLUZIONE:

punto 1

Data la configurazione iniziale del vettore, per poter costruire uno heap è necessario applicare la funzione `Build-heap`. Inoltre, tenendo conto che nella radice deve essere memorizzato l'elemento del vettore di valore minimo, è necessario modificare la funzione `Heapify(i)` in modo che ad ogni passo l'elemento di valore minimo tra `i`, il suo figlio di sinistra `l`, ed il suo figlio di destra `r`, venga scambiato con `i`.

I passi della costruzione dello heap sono i seguenti:

	1	2	3	4	5	6	7	8	9	10	
Heapify(5)	32	45	55	61	21	29	39	38	37	33	Nessun cambiamento
Heapify(4)	32	45	55	61	21	29	39	38	37	33	Scambia 61 con 37
Heapify(3)	32	45	55	37	21	29	39	38	61	33	Scambia 55 con 29
Heapify(2)	32	45	29	37	21	55	39	38	61	33	Scambia 45 con 21
Heapify(5)	32	21	29	37	45	55	39	38	61	33	Scambia 45 e 33
Heapify(1)	32	21	29	37	33	55	39	38	61	45	Scambia 32 e 21
Fine	21	32	29	37	33	55	39	38	61	45	

Dato lo heap appena costruito, i primi due passi dello heapsort sono i seguenti:

	1	2	3	4	5	6	7	8	9	10	
Swap(1, 10)	21	32	29	37	33	55	39	38	61	45	Scambia 21 con 45
Heapify(1)	45	32	29	37	33	55	39	38	61	21	Scambia 45 con 29
Heapify(3)	29	32	45	37	33	55	39	38	61	21	Scambia 45 con 39
Swap(1, 9)	29	32	39	37	33	55	45	38	61	21	Scambia 29 con 61
Heapify(1)	61	32	39	37	33	55	45	38	29	21	Scambia 61 con 32
Heapify(2)	32	61	39	37	33	55	45	38	29	21	Scambia 61 con 33
	32	33	39	37	61	55	45	38	29	21	

punto 2

	1	2	3	4	5	6	7	8	9	10	
Partition(1,10)	32	45	55	61	21	29	39	38	37	33	Pivot = 32, scambia 32 con 29
	29	45	55	61	21	32	39	38	37	33	Pivot = 32, scambia 45 con 21
	29	21	55	61	45	32	39	38	37	33	Partition(1, 10) ritorna 2
Partition(1,2)	29	21									Pivot = 29, scambia 29 con 21
	21	29									Partition(1, 2) ritorna 1
Partition(3, 10)			55	61	45	32	39	38	37	33	Pivot = 55, scambia 55 con 33
			33	61	45	32	39	38	37	55	Pivot = 55, scambia 61 con 37
			33	37	45	32	39	38	61	55	Partition(3, 10) ritorna 8

In conclusione, si ha che al primo passo le due partizioni sono $P1=\{29, 21\}$ e $P2=\{55, 61, 45, 32, 39, 38, 37, 33\}$. Al secondo passo, le due partizioni di P1 sono $P11=\{21\}$, $P12=\{29\}$ e le due partizioni di P2 sono $P21=\{33, 37, 45, 32, 39, 38\}$, $P22=\{61, 55\}$.

Esercizio 2

Sia data la sequenza di chiavi AUTUNNO, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto (A=1, ..., Z=26). Si riporti la struttura di una tabella di hash di dimensione 19, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza di cui sopra. Si supponga di utilizzare l'open addressing con probing lineare scegliendo opportunamente la funzione di hash.

SOLUZIONE:

La funzione di hash utilizzata è la seguente:

$$H(k) = \text{ordine}(k) \bmod 19$$

Dove la funzione ordine(k) ritorna la posizione nell'alfabeto del carattere k, assunto l'ordine progressivo indicato nel testo.

I valori dalla funzione di hash per le chiavi indicate nel testo sono i seguenti:

$$H(A) = 1$$

$$H(U) = 2$$

$$H(T) = 1$$

$$H(N) = 14$$

$$H(O) = 15$$

Il contenuto della tabella di hash in seguito all'inserimento delle chiavi indicate sarà pertanto:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	A	U	T	U										N	N	O		

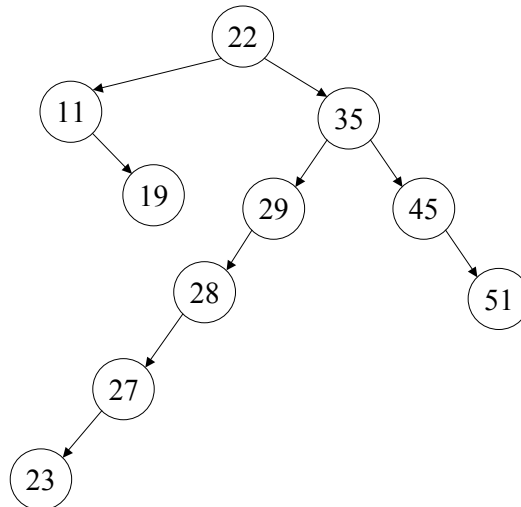
Esercizio 3

Si effettuino secondo l'ordine specificato le seguenti operazioni su un BST supposto inizialmente vuoto (+indica una inserzione, - una cancellazione):

+22 +35 +45 +51 +11 +19 +29 +28 +27 +23 -35 -19

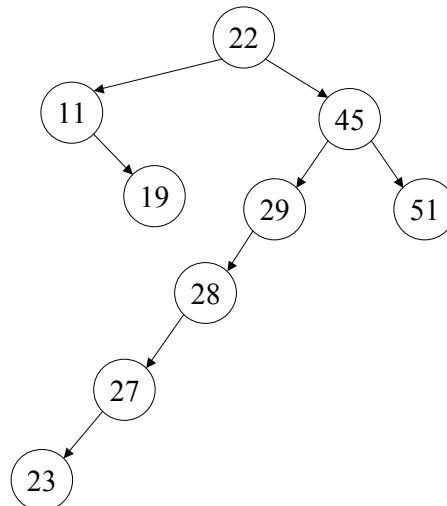
SOLUZIONE:

Al termine della fase di inserimento il contenuto del BST è il seguente:

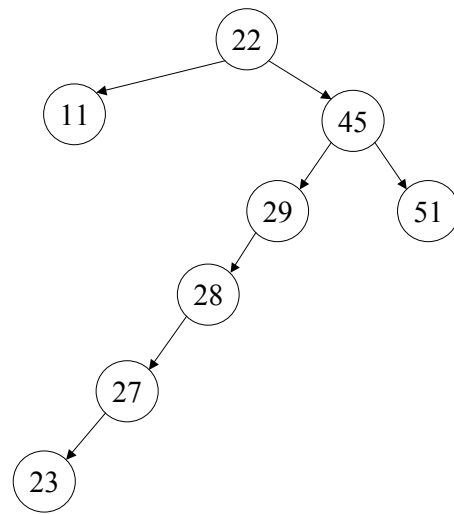


La cancellazione del nodo 35 deve essere effettuata utilizzando l'algoritmo di cancellazione da un BST di un nodo con 2 figli. È quindi necessario identificare il successore di 35, cancellarlo e sostituirlo al posto di 35. In questo caso, il successore di 35 è la chiave minima nel suo sottoalbero di destra: 45.

Dopo la cancellazione della chiave 35, il contenuto del BST è quindi:



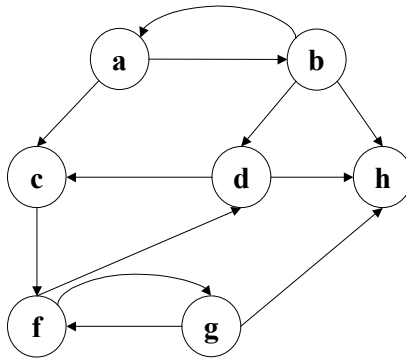
La cancellazione del nodo 19 richiede invece l'utilizzo dell'algoritmo di cancellazione di un nodo che non ha figli: è quindi sufficiente cancellare il nodo, ottenendo il seguente BST:



Esercizio 4

Sia dato il grafo orientato in figura:

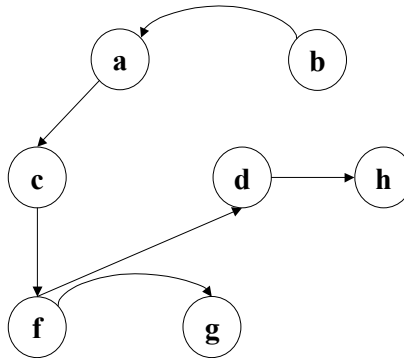
1. Se ne effettui una visita in profondità, considerando **b** come vertice di partenza. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.
2. Se ne effettui la visita in ampiezza, considerando **b** come vertice di partenza. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.



SOLUZIONE:

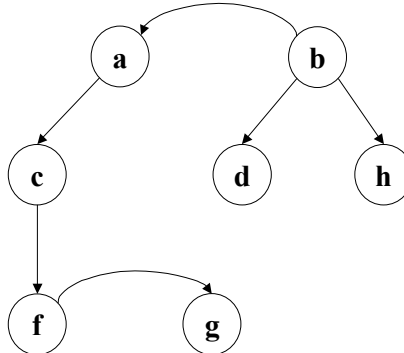
punto 1

Effettuando la visita in profondità a partire dal nodo **b** si ottiene il seguente risultato:



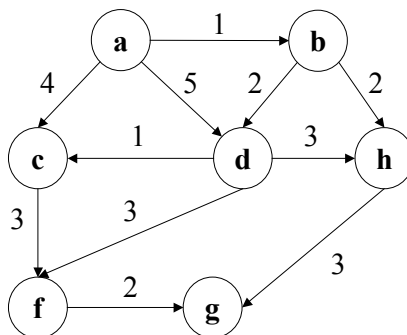
punto 2

Effettuando la visita in ampiezza a partire dal nodo **b** si ottiene il seguente risultato:



Esercizio 5

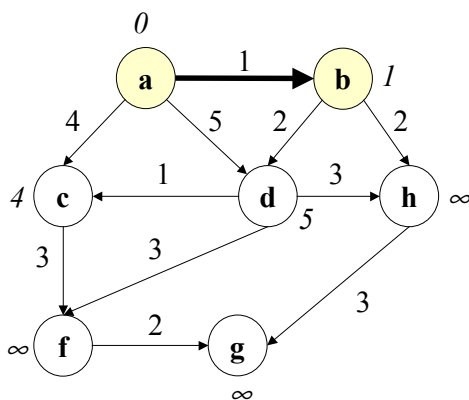
Sia dato il seguente grafo orientato pesato. Si determinino i valori di tutti i cammini minimi che collegano il vertice **a** con ogni altro vertice mediante l'algoritmo di Dijkstra. Si assuma, qualora necessario, un ordine alfabetico per i vertici e gli archi.



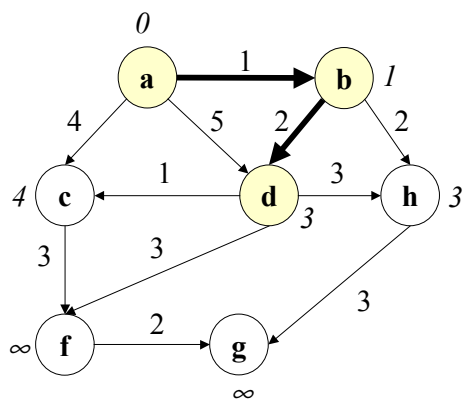
SOLUZIONE:

Le figure seguenti riportano il risultato dei passi dell'algoritmo di Dijkstra. A fianco di ogni nodo è riportato in *italico* il costo associato al nodo.

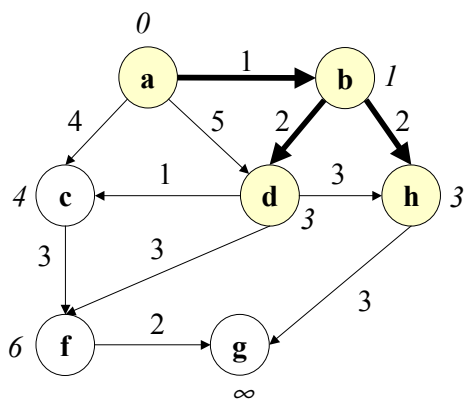
Passo 1



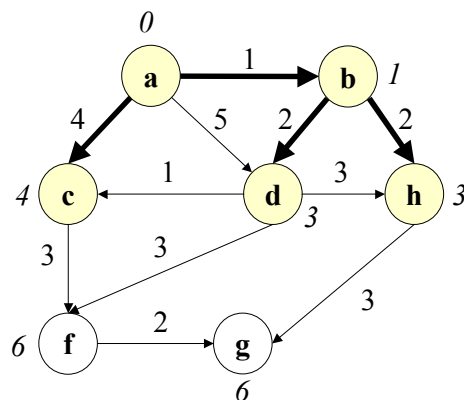
Passo 2



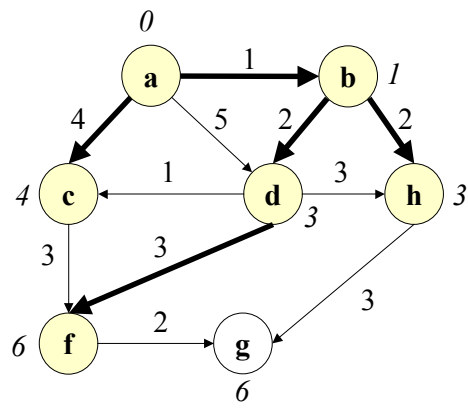
Passo 3



Passo 4



Passo 5



Passo 6

