



Esercizi SL4



Esercizio

Enunciare un teorema che stabilisca il legame tra numero delle foglie e numero dei nodi interni di un albero ternario pieno (analogo al *full binary tree theorem* studiato per gli alberi binari) e dimostrarlo per induzione.



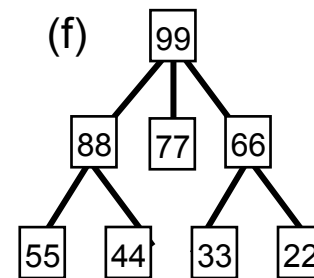
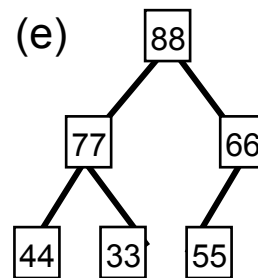
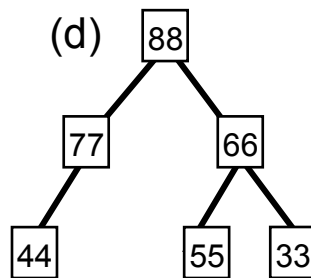
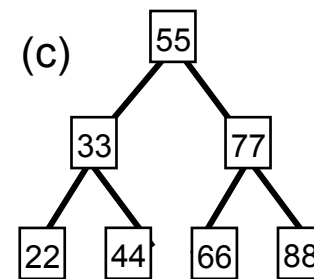
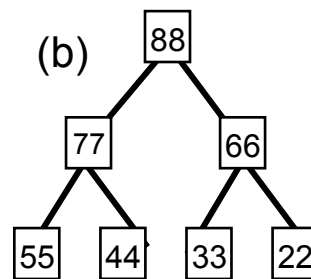
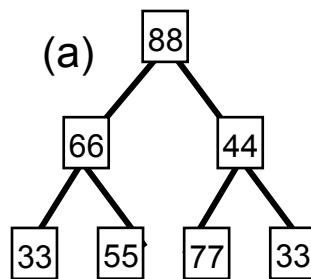
Soluzione

- Enunciato: In un albero ternario pieno con n nodi interni il numero delle foglie è uguale a $2n+1$.

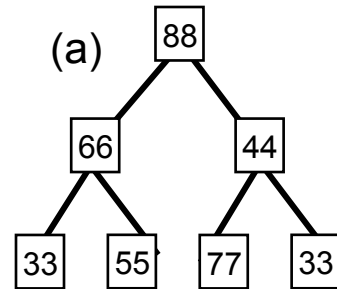
- Dimostrazione:
 - Caso base: un albero con 0 nodi interni ha 1 foglia.
 - Induzione. A una foglia qualsiasi di un albero con k nodi interni e $2k+1$ foglie attacco tre foglie: ora ho un nodo interno in più e due foglie in più (ne tolgo una e ne aggiungo tre): il numero dei nodi interni è $k+1$ e il numero delle foglie è $(2k+1)-1+3 = 2k+3 = 2(k+1)+1$.

Esercizio

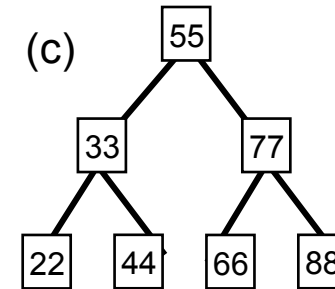
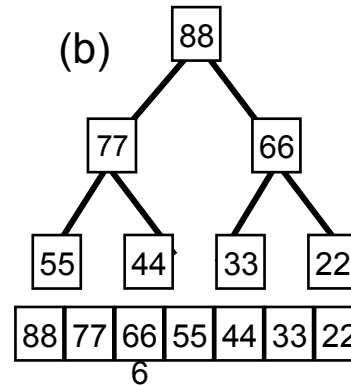
Quale dei seguenti alberi è uno heap? Per i casi negativi indicare il motivo, per quelli positivi disegnare un array che lo memorizzi.



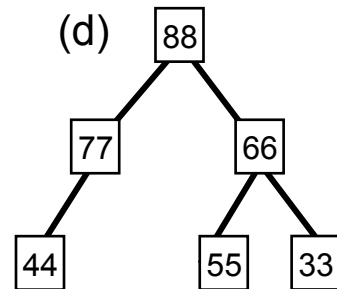
Soluzione



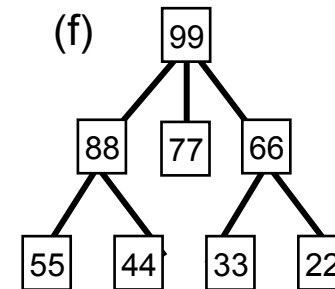
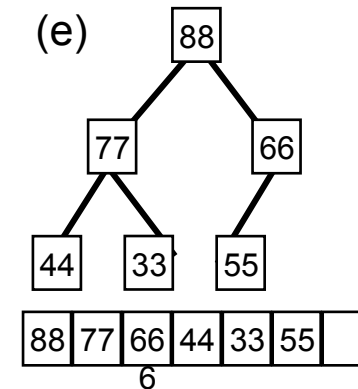
NO: 88-44-77



NO: 55-33-44; 55-77-88



NO: albero non completo



NO: albero non binario

Esercizio

Quale dei seguenti array memorizza uno heap? Per i casi negativi, indicare il motivo, per quelli positivi disegnare lo heap corrispondente.

(a)

88	66	44	33	55	77	33
----	----	----	----	----	----	----

(b)

88	77	66	55	44	33	22
----	----	----	----	----	----	----

(c)

88	44	77	22	33	55	66
----	----	----	----	----	----	----

(d)

88	77	55	44		33	22
----	----	----	----	--	----	----

(e)

88	66	77	22	33	44	55
----	----	----	----	----	----	----

(f)

88	77	22	33	44	55	66
----	----	----	----	----	----	----

Soluzione

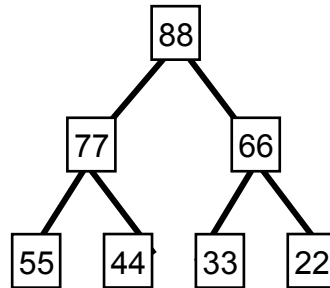
(a)

88	66	44	33	55	77	33
----	----	----	----	----	----	----

NO: 88-44-55

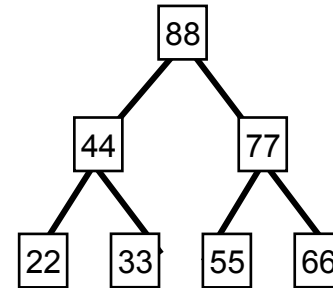
(b)

88	77	66	55	44	33	22
----	----	----	----	----	----	----



(c)

88	44	77	22	33	55	66
----	----	----	----	----	----	----



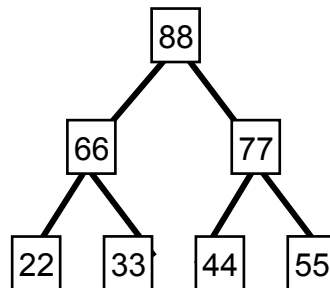
(d)

88	77	55	44		33	22
----	----	----	----	--	----	----

NO: albero non completo

(e)

88	66	77	22	33	44	55
----	----	----	----	----	----	----



(f)

88	77	22	33	44	55	66
----	----	----	----	----	----	----

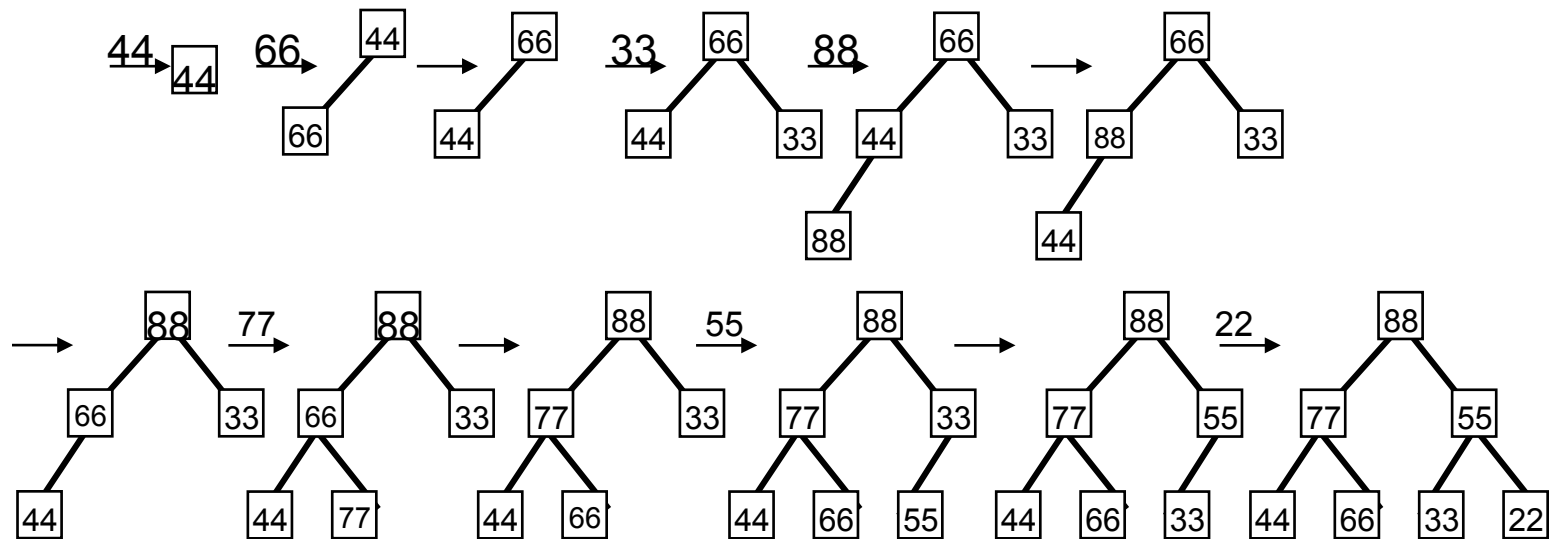
NO: 88-22-55; 88-22-66



Esercizio

Visualizzare la costruzione dello heap derivante dall'inserzione dei seguenti valori: 44, 66, 33, 88, 77, 55, 22. Mostrare anche le operazioni di ristrutturazione dell'albero durante le inserzioni.

Soluzione





Esercizio 5

Data la seguente lista, applicare
l'ordinamento di mergesort passo passo:

42 16 28 36 26 78 84 8



Soluzione

42 16 28 36 26 78 84 8

16	42	28	36	26	78	8	84
----	----	----	----	----	----	---	----

16	28	36	42	8	26	78	84
----	----	----	----	---	----	----	----

8	16	26	28	36	42	78	84
---	----	----	----	----	----	----	----



16	28	36	42
----	----	----	----

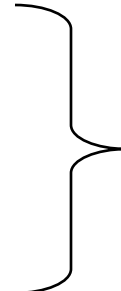
8	26	78	84
---	----	----	----

16	28	36	42
----	----	----	----

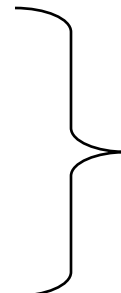
26	78	84
----	----	----

28	36	42
----	----	----

26	78	84
----	----	----



--



8



8 16



28 36 42

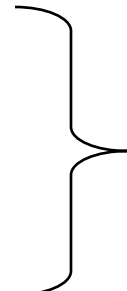
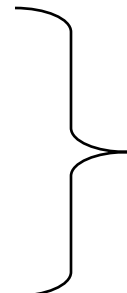
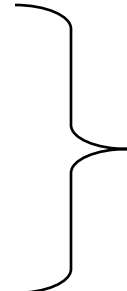
78 84

36 42

78 84

42

78 84



8 16 26

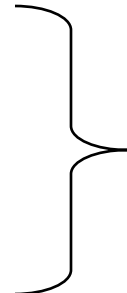
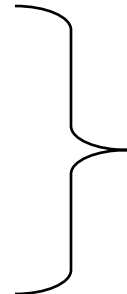
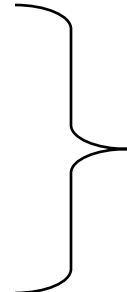
8 16 26 28

8 16 26 28 36



78 84

84



8 16 26 28 36 42

8 16 26 28 36 42 78

8 16 26 28 36 42 78 84



Hashing

- Data una tabella hash di lunghezza $m=11$, si supponga di dover inserire (in ordine) le chiavi: 35, 83, 57, 26, 15, 63, 97, 46, con la funzione di hash $h(k) = k \bmod m$. Si illustrino i risultati dell'inserimento usando:
 - ☐ separate chaining
 - ☐ linear probing
 - ☐ quadratic probing ($h_i(k) = (h(k) + i^2) \bmod m$)
 - ☐ double hashing con $h_2(K) = 1 + (k \bmod (m-1))$

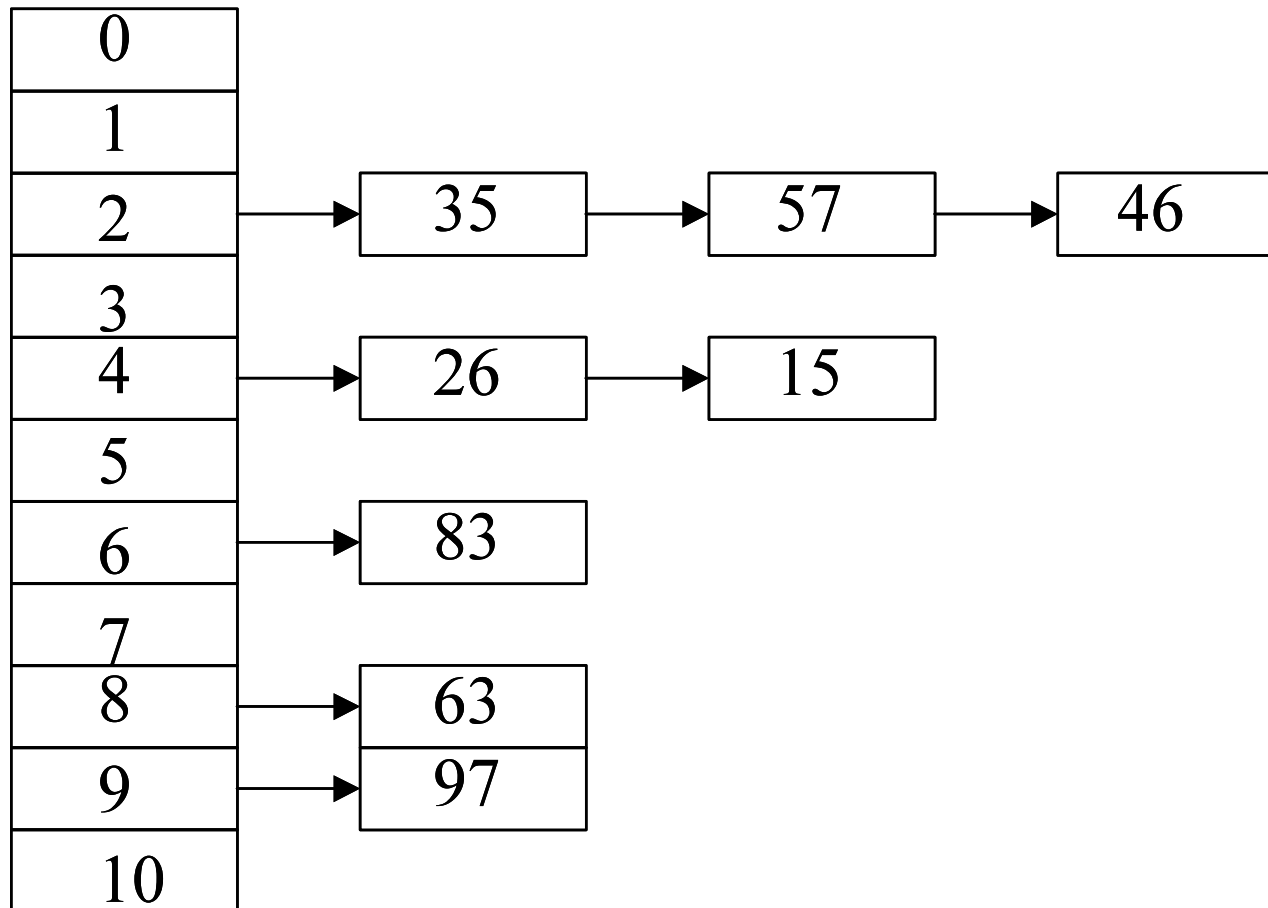


Soluzione

■ Calcolo di $h(k)$

- ☐ $h(35) = 35 \bmod 11 = 2$
- ☐ $h(83) = 83 \bmod 11 = 6$
- ☐ $h(57) = 57 \bmod 11 = 2$
- ☐ $h(26) = 26 \bmod 11 = 4$
- ☐ $h(15) = 15 \bmod 11 = 4$
- ☐ $h(63) = 63 \bmod 11 = 8$
- ☐ $h(97) = 97 \bmod 11 = 9$
- ☐ $h(46) = 46 \bmod 11 = 2$

Separate chaining



$h(35)=2$
 $h(83)=6$
 $h(57)=2$
 $h(26)=4$
 $h(15)=4$
 $h(63)=8$
 $h(97)=9$
 $h(46)=2$

Linear probing

0	1	2	3	4	5	6	7	8	9	10
		35	57	26	15	83	46	63	97	

$h(57)=2 \rightarrow$ lo slot 2 è occupato $h_1(57)=3$

$h(15)=4 \rightarrow$ lo slot 4 è occupato $h_1(15)=5$

$h(46)=2 \rightarrow$ lo slot 2 è occupato

$h_1(46)=3 \rightarrow$ lo slot 3 è occupato

$h_2(46)=4 \rightarrow$ lo slot 4 è occupato

$h_3(46)=5 \rightarrow$ lo slot 5 è occupato

$h_4(46)=6 \rightarrow$ lo slot 6 è occupato $h_5(46)=7$

$h(35)=2$

$h(83)=6$

$h(57)=2$

$h(26)=4$

$h(15)=4$

$h(63)=8$

$h(97)=9$

$h(46)=2$



Quadratic probing $h_i(k) = (h(k) + i^2) \bmod m$

0	1	2	3	4	5	6	7	8	9	10
46		35	57	26	15	83		63	97	

$h(57)=2 \rightarrow$ lo slot 2 è occupato $h_1(57)=3$

$h(15)=4 \rightarrow$ lo slot 4 è occupato $h_1(15)=5$

$h(46)=2 \rightarrow$ lo slot 2 è occupato

$h_1(46)=3 \rightarrow$ lo slot 3 è occupato

$h_2(46)=6 \rightarrow$ lo slot 6 è occupato $h_3(46)=0$

$h(35)=2$
$h(83)=6$
$h(57)=2$
$h(26)=4$
$h(15)=4$
$h(63)=8$
$h(97)=9$
$h(46)=2$

Double hashing

$$h_2(K)=1+(k \bmod (m-1)) \rightarrow h_i(k)=(h_1(k)+i*h_2(k)) \bmod m$$

0	1	2	3	4	5	6	7	8	9	10
	46	35		26	15	83		63	97	57

$h(57)=2 \rightarrow$ lo slot 2 è occupato $h_1(57)=2+1*8=10$

$h(15)=4 \rightarrow$ lo slot 4 è occupato $h_1(15)=4+1*6=10$

\rightarrow lo slot 10 è occupato $h_2(15)=4+2*6=5$

$h(46)=2 \rightarrow$ lo slot 2 è occupato $h_1(46)=2+1*7=9$

\rightarrow lo slot 9 è occupato $h_2(46)=2+2*7=5$

\rightarrow lo slot 5 è occupato $h_3(46)=2+3*7=1$

$h(35)=2$
$h(83)=6$
$h(57)=2$
$h(26)=4$
$h(15)=4$
$h(63)=8$
$h(97)=9$
$h(46)=2$



Algoritmi

- Nel gioco del SuGiu un giocatore deve indovinare un numero, compreso in un intervallo prefissato, attraverso una serie di tentativi. A ogni tentativo il giocatore indica un valore e un interlocutore che conosce il numero da indovinare dà una delle seguenti tre risposte: SI se il numero è stato indovinato, SU se il numero da indovinare dal giocatore è *superiore* al numero indicato, GIU se è *inferiore*. Tratteggiare l'implementazione di un programma che, ricevendo come parametri in ingresso gli estremi inferiore e superiore dell'intervallo di possibili valori per il numero da indovinare, riesca a trovarlo con un numero di tentativi minimo (si assuma che il programma possa fare la parte del giocatore, scrivendo sull'output standard i suoi tentativi e leggendo le indicazioni dell'interlocutore dall'input standard).



Soluzione

Bozza di programma:

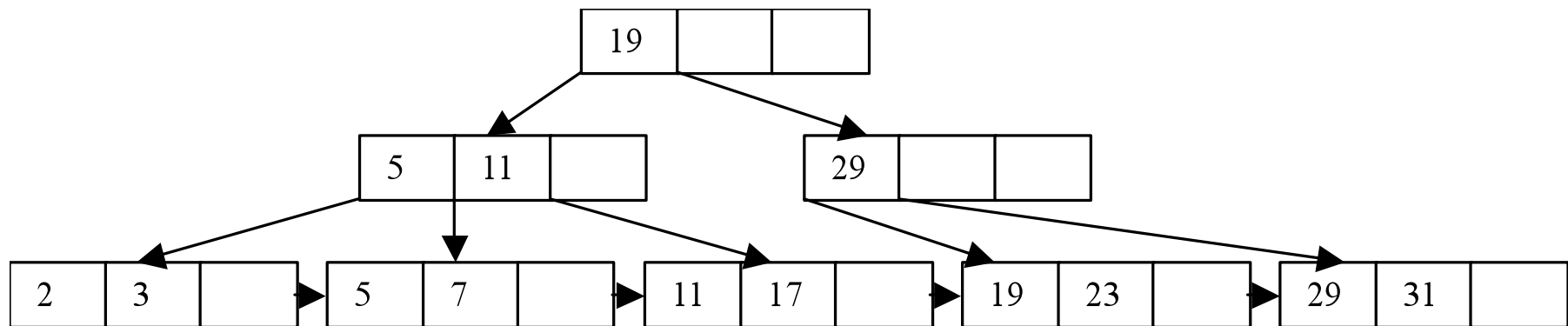
```
static void suGiu (int inf, int sup) {  
    int med;  
    String risposta;  
    do {  
        med = (inf + sup) / 2;  
        scrivi med;  
        leggi risposta;  
        if (risposta = GIU)  
            sup = med - 1;  
        else if (risposta = SU)  
            inf = med + 1;  
    } while (risposta != SI);  
}
```

Strategia: simulare una
ricerca binaria

Complessità: logaritmica
rispetto all'ampiezza
dell'intervallo di valori
ammissibili

B+ tree

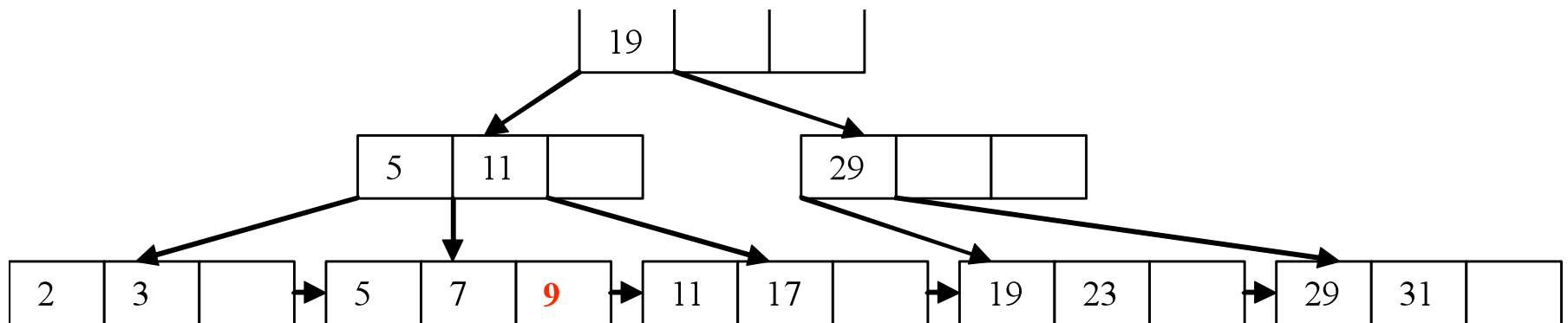
- Si consideri il seguente albero:



- a) Di che tipo e ordine è?
- b) Si eseguano in sequenza le seguenti operazioni:
- Inserisci 9, Inserisci 10, Inserisci 8, Cancella 23, Cancella 19.

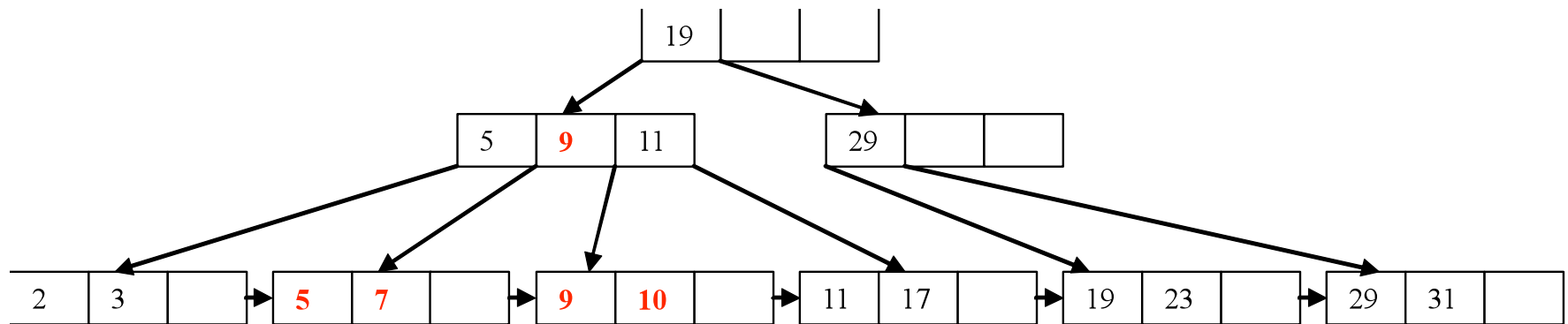
Soluzione

- a) B+, $m=4$
- b) *Inserisci 9:*



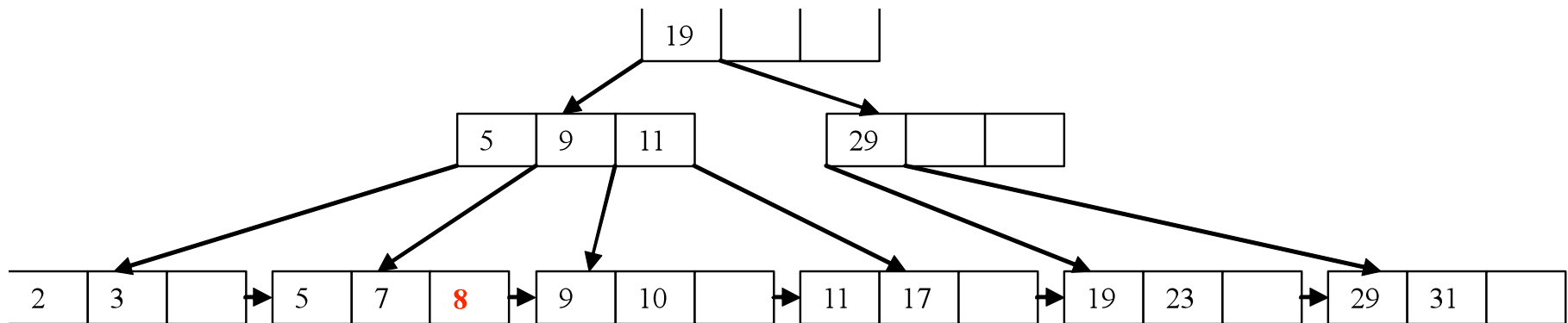
Soluzione

■ *Inserisci 10:*



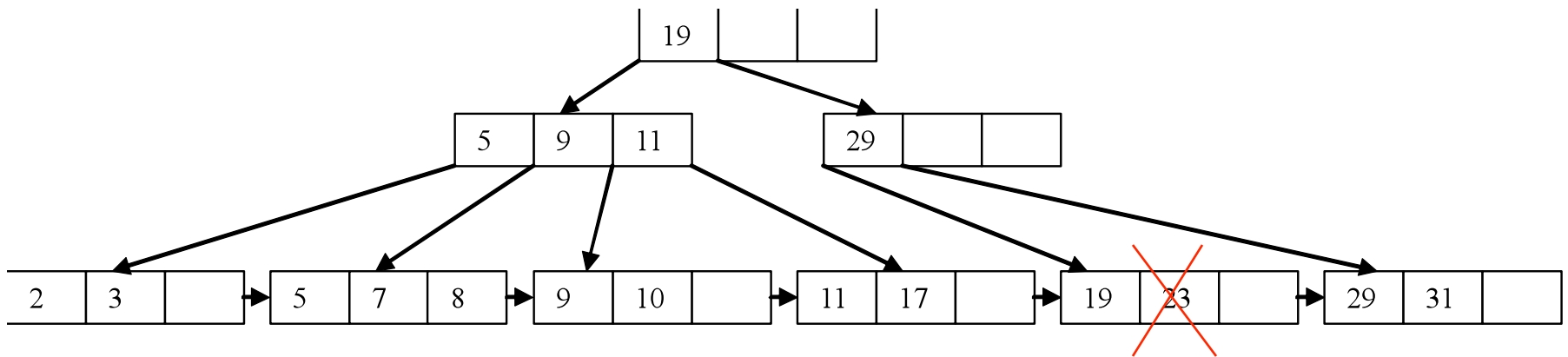
Soluzione

■ *Inserisci 8:*



Soluzione

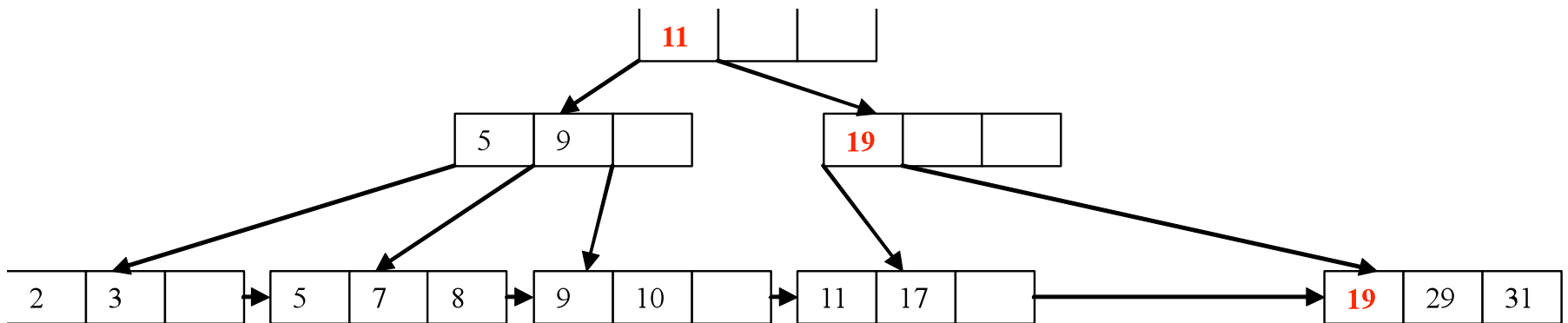
■ *Cancella 23:*



Underflow!!!

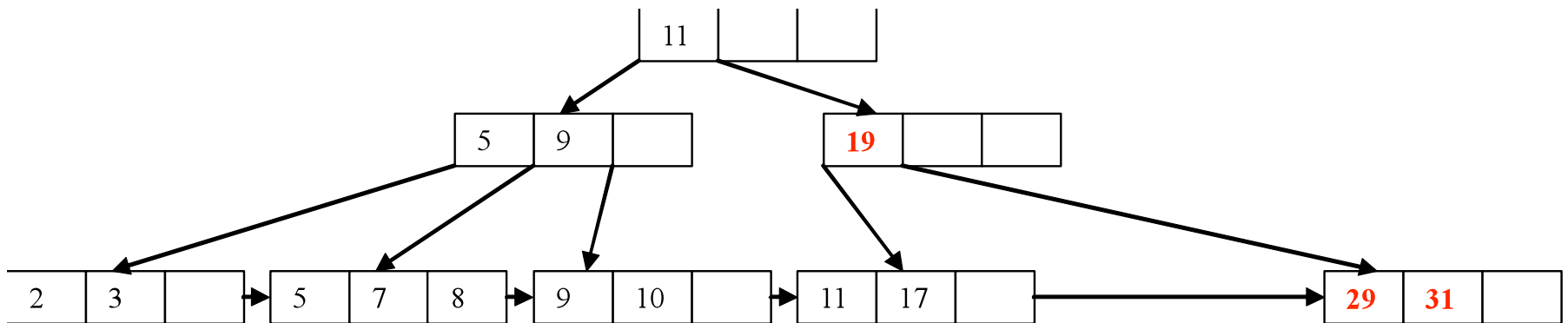
Soluzione

■ *Cancella 23:*



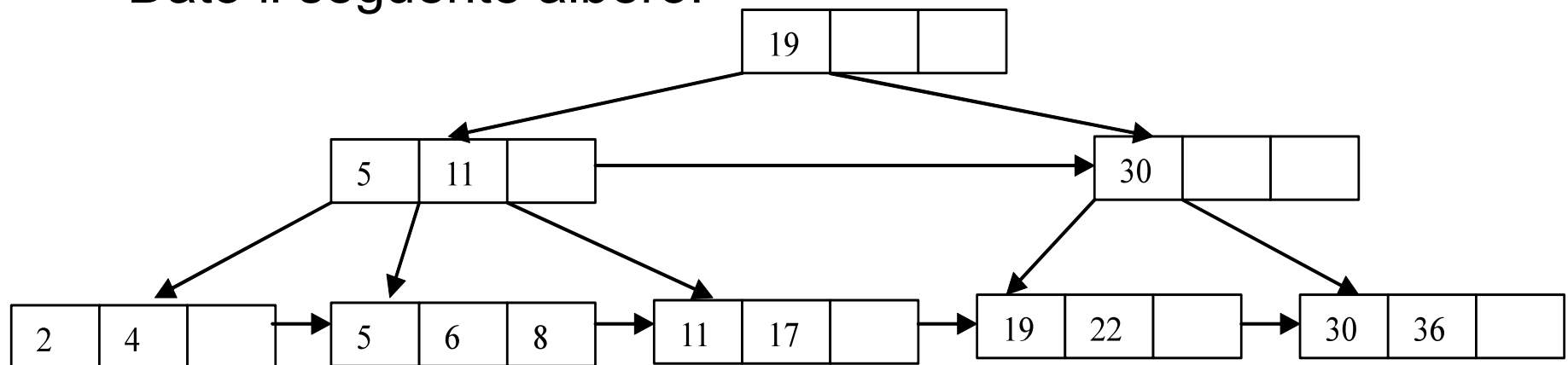
Soluzione

■ *Cancella 19:*



B+ tree

- Dato il seguente albero:



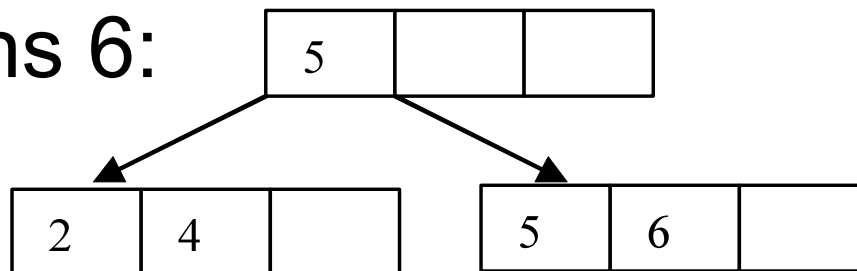
- a) Di che tipo e ordine è?
- b) Si indichi una sequenza di inserimenti che, partendo dall'albero vuoto, porta all'albero indicato, mostrando gli alberi intermedi ottenuti a seguito delle operazioni di split.

Soluzione

- a) B+ di ordine 4
- b) ins 2, ins 4, ins 5:

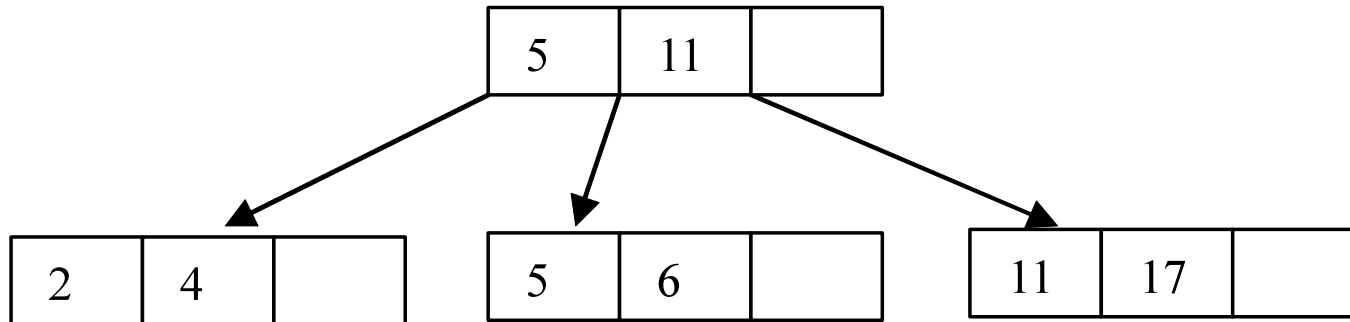


- ins 6:

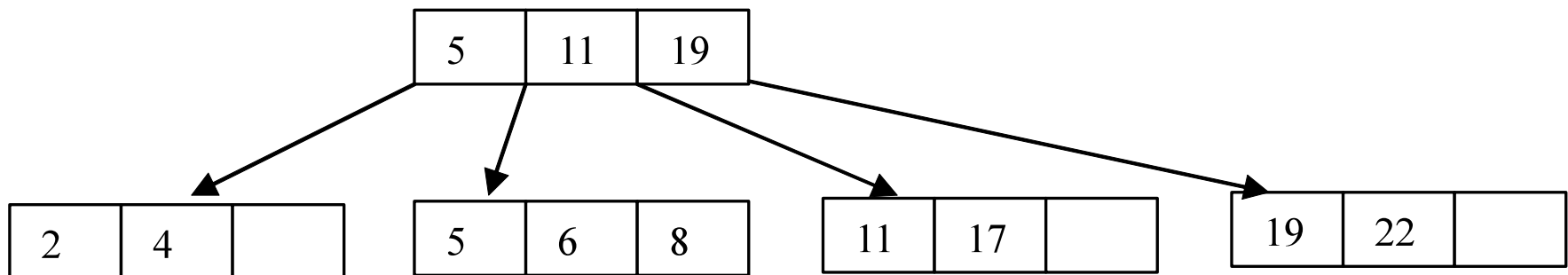


Soluzione

■ ins 11, ins 17:

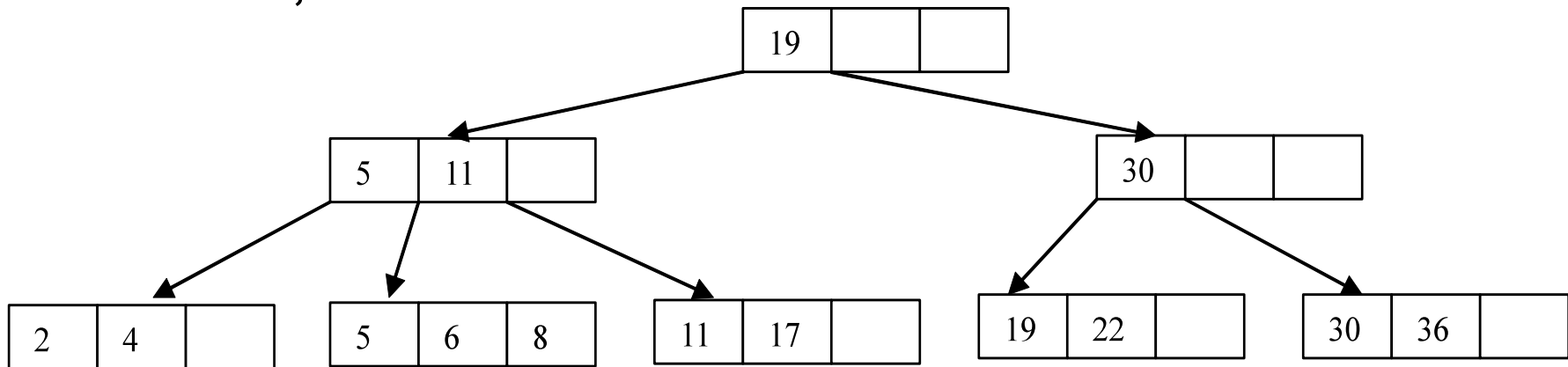


■ ins 8, ins 19, ins 22:



Soluzione

- ins 30, ins 36:



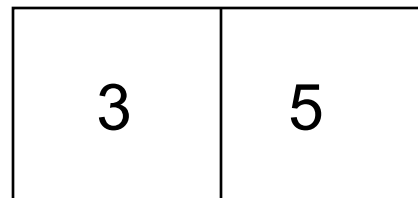
- N.B.
- la soluzione non è univoca: ad es. si può inserire 8 in una qualsiasi fase successiva a quella mostrata (ma non precedente), oppure permutare alcuni blocchi di inserimenti (come 2 e 4).



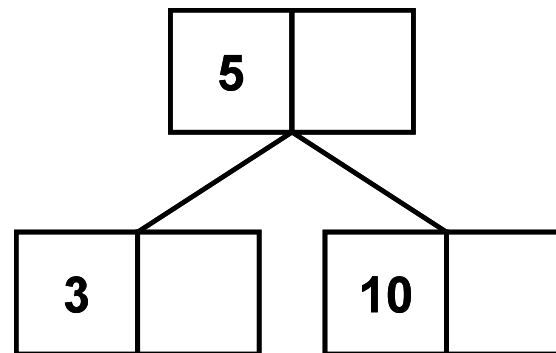
Albero 2-3

- Si consideri la seguente sequenza di record le cui chiavi sono nell'ordine:
3, 5, 10, 6, 8, 14, 7, 15, 9, 11, 13, 12
- Si mostri l'albero 2-3 risultante dagli inserimenti di questi record.

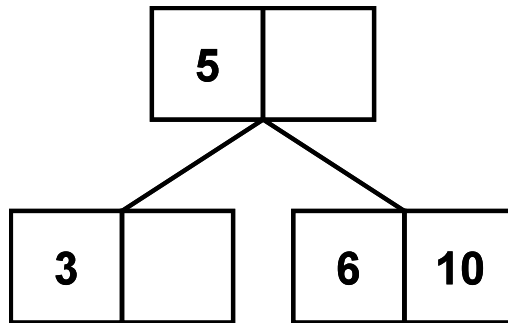
- La realizzazione del 2-3 tree inizia con l'inserimento dei valori 3 e 5.



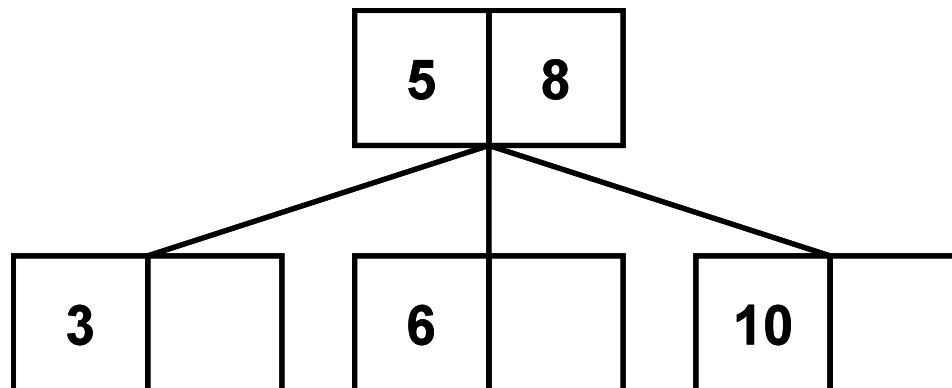
- Inserimento del valore 10



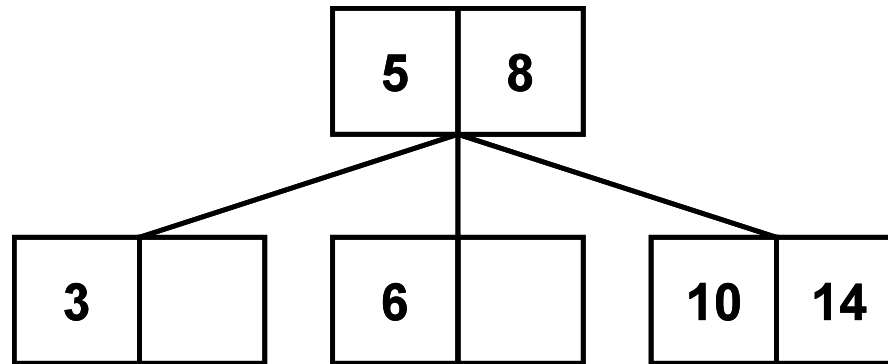
■ Inserimento del valore 6



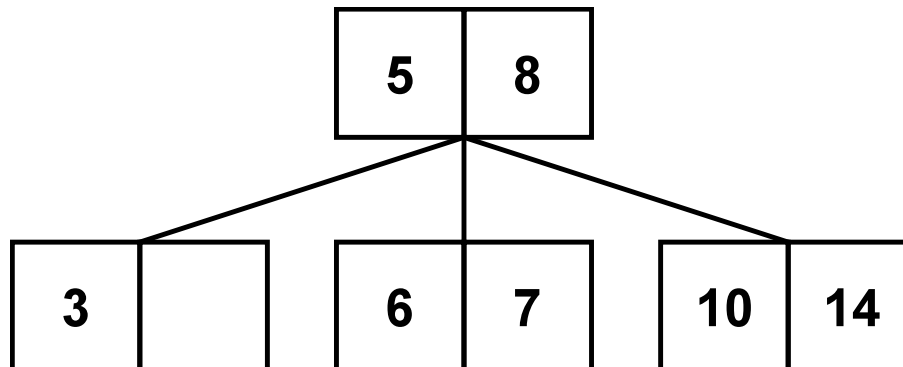
■ Inserimento del valore 8



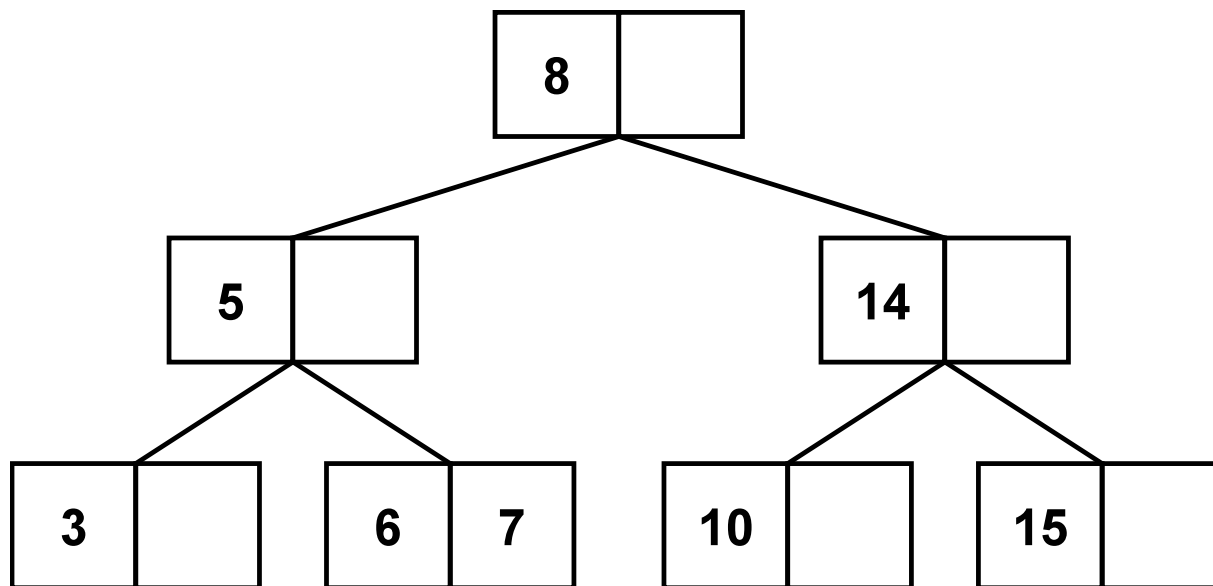
■ Inserimento del valore 14



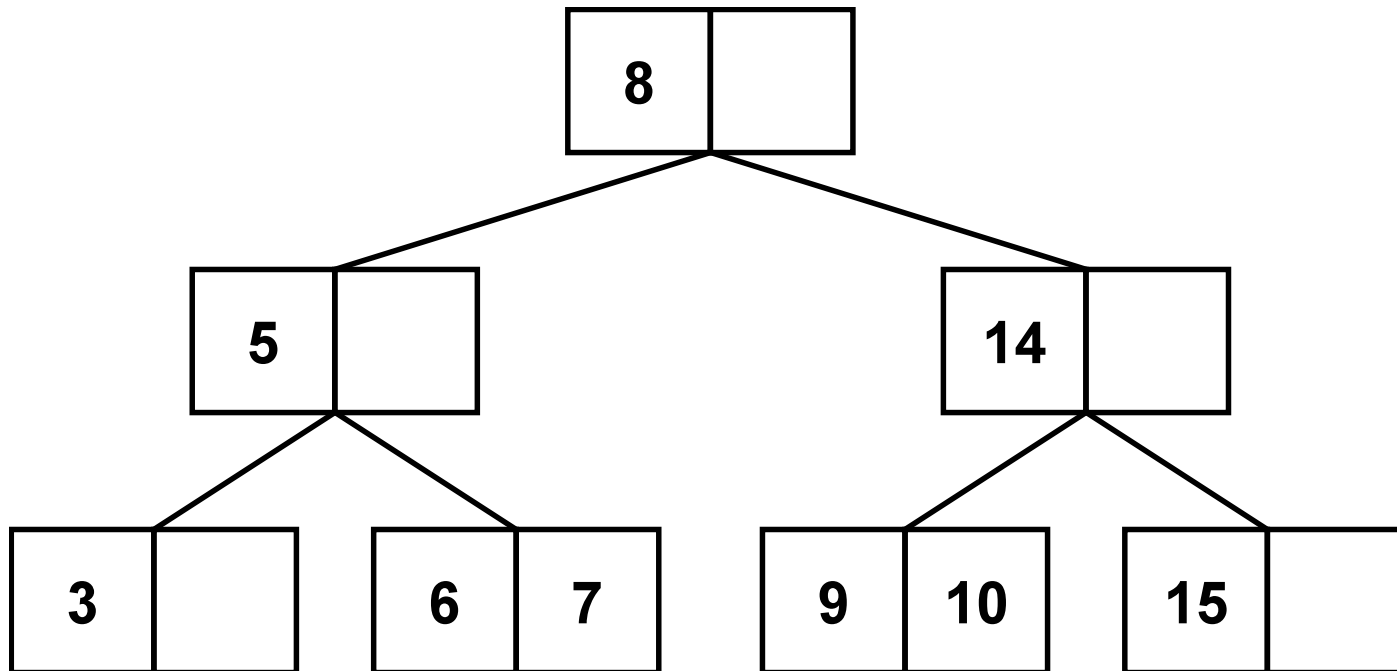
■ Inserimento del valore 7



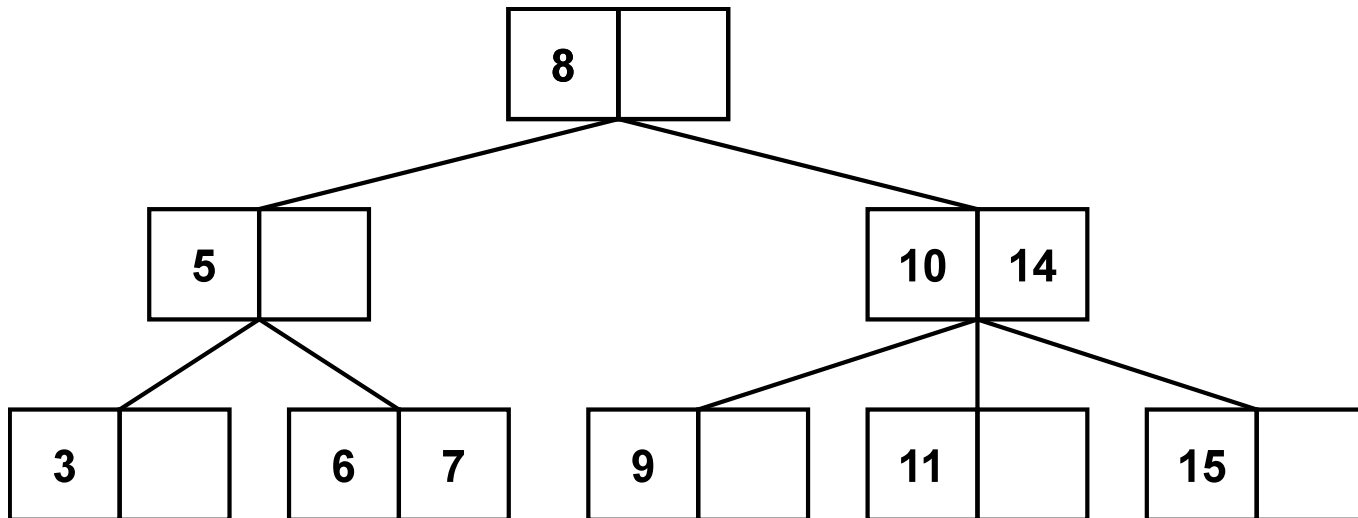
■ Inserimento del valore 15



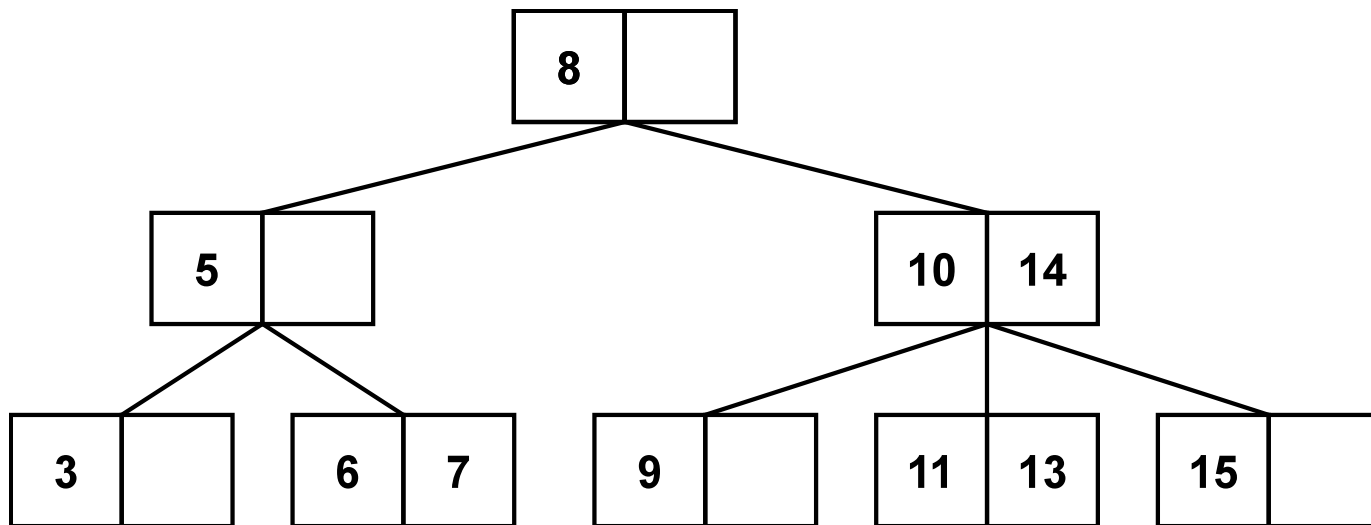
■ Inserimento del valore 9



■ Inserimento del valore 11



■ Inserimento del valore 13



■ Inserimento del valore 12

