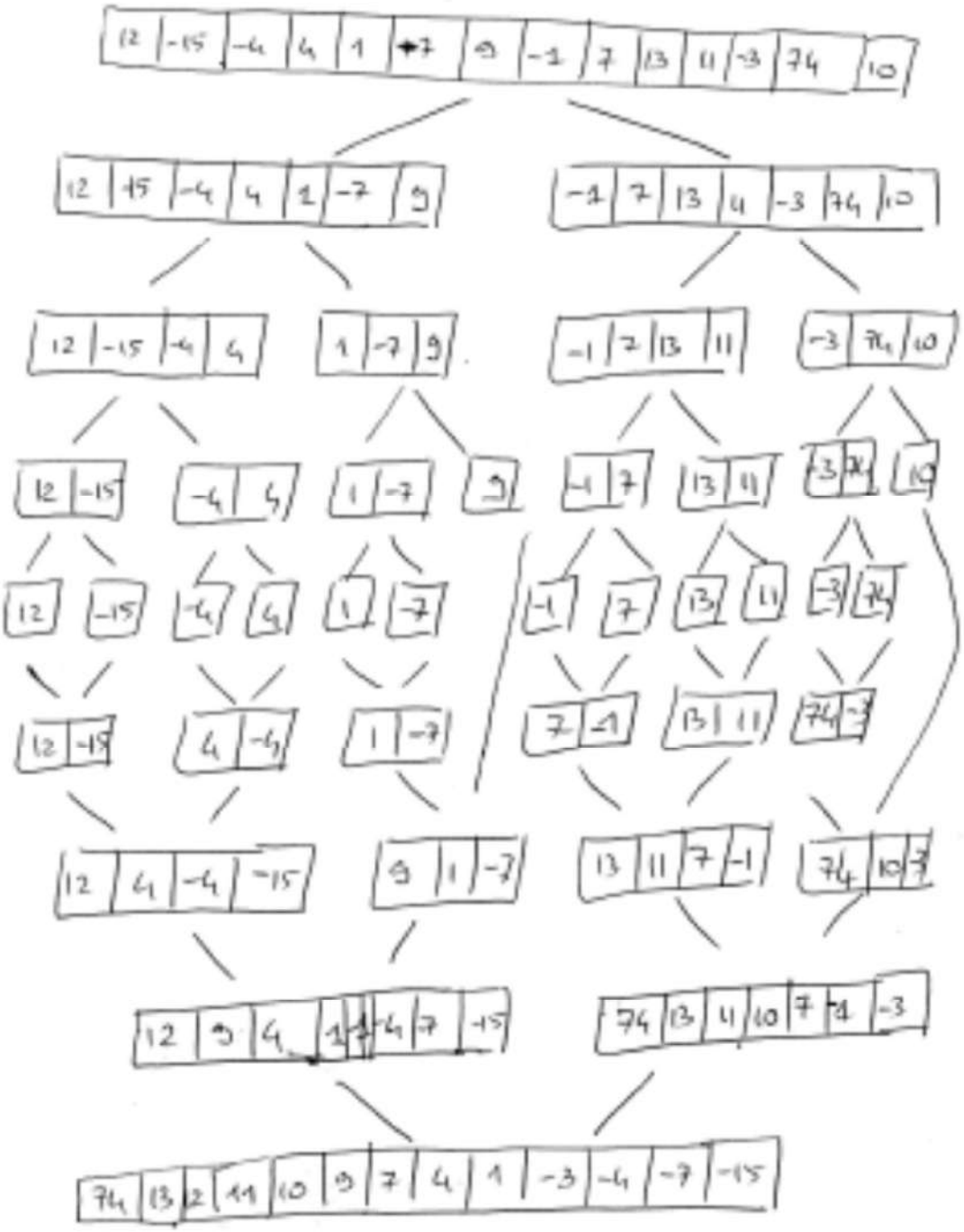
Merge-sort

2.1.3 Esercizio proposto pag. 25

Si ordini in maniera discendente la seguente sequenza di interi mediante merge sort.

12 -15 -4 4 1 -7 9 -1 7 13 11 -3 74 10

Si indichino i passaggi principali.


12 -15 -4 4 1 -7 9 -1 7 13 11 -3 74 10
12 -15 -4 4 1 -7 9
-1 7 13 11 -3 74 10
12 -15 -4 4
1 -7 9
-1 7 13 11
-3 74 10
12 -15
-4 4
1 -7
9
-1 7
13 11
-3 74
10
12
-15
-4
4
1
-7
-1
7
13
11
-3
74
12 -15
4 -4
1 -7
7 -1
13 11
74 -3
12 4 -4 -15
9 1 -7
13 11 7 -1
74 10 -3
12 9 4 1 -4 -7 -15
74 13 4 10 7 1 -3
74 13 12 11 10 9 7 4 1 -3 -4 -7 -15

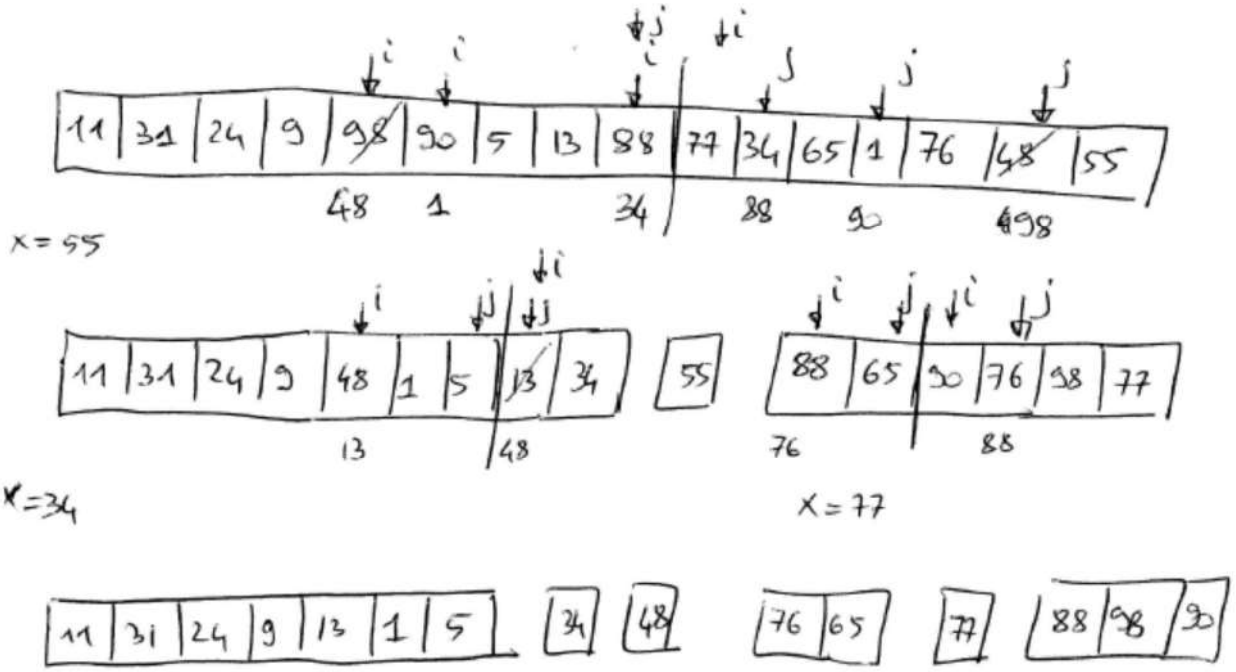
Quick-sort

2.2.4 Esercizio proposto pag. 34

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

11 31 24 9 98 90 5 13 88 77 34 65 1 76 48 55

si eseguano i primi due passi dell'algoritmo di quick sort per ottenere un ordinamento ascendente, indicando ogni volta il pivot scelto. Si osservi che i passi sono da intendersi, impropriamente, come in ampiezza sull'albero della ricorsione, non in profondità. Si chiede, pertanto, che siano ritornate le due partizioni del vettore originale e le due partizioni delle partizioni trovate al punto precedente. Si utilizzi la tecnica di partizionamento di Hoare.

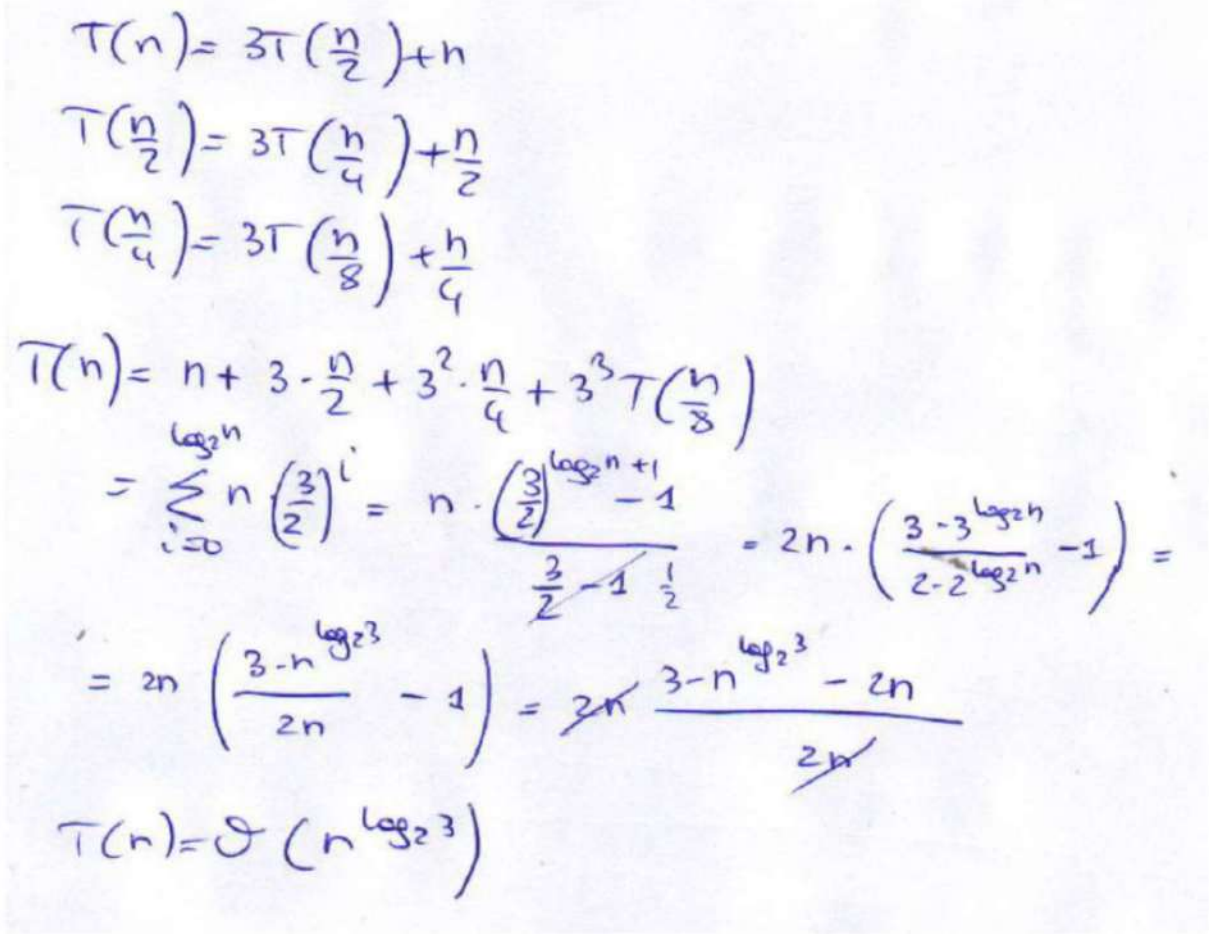


Equazioni alla ricorrenze

3.1.3 Esercizio risolto pag. 39

Si risolva, mediante il metodo dello sviluppo (unfolding), la seguente equazione alle ricorrenze:

$$T(n) = 3 \cdot T(\tfrac{n}{2}) + n \qquad n \geq 2$$
$$T(1) = 1$$



3.1.3 Esercizio proposto pag. 41

Si risolva mediante il metodo dello sviluppo (unfolding) la seguente equazione alle ricorrenze:

$$\begin{aligned} T(n) &= 2 \cdot T(\tfrac{n}{3}) + n^2 \qquad n \geq 2 \\ T(1) &= 1 \end{aligned}$$

$$T(n) = 2\,T\left(\frac{n}{3}\right) + n^2$$

$$T\left(\frac{n}{3}\right) = 2\,T\left(\frac{n}{9}\right) + \left(\frac{n}{3}\right)^2$$

$$T\left(\frac{n}{9}\right) = 2\,T\left(\frac{n}{27}\right) + \left(\frac{n}{9}\right)^2$$

sostituisco $T\left(\frac{n}{9}\right)$ in $T\left(\frac{n}{3}\right)$

$$T\left(\frac{n}{3}\right) = \left(\frac{n}{3}\right)^2 + 2\cdot\left(\frac{n}{9}\right)^2 + 2^2\,T\left(\frac{n}{27}\right)$$

sostituisco $T\left(\frac{n}{3}\right)$ in $T(n)$

$$T(n) = n^2 + 2\cdot\left(\frac{n}{3}\right)^2 + 2^2\left(\frac{n}{9}\right)^2 + 2^3\,T\left(\frac{n}{27}\right)$$

$$T(n) = \sum_{i=0}^{?} 2^i \cdot \left(\frac{n}{3^i}\right)^2 \qquad ? \quad \frac{n}{3^i} = 1 \qquad n = 3^i \qquad i = \log_3 n$$

$$T(n) = \sum_{i=0}^{\log_3 n} 2^i \cdot \frac{n^2}{3^{2i}} = n^2 \cdot \sum_{i=0}^{\log_3 n} \left(\frac{2}{9}\right)^i$$

$$\sum_{i=0}^{k} x^k = \frac{x^{k+1} - 1}{x - 1}$$

$$= n^2 \cdot \frac{\left(\frac{2}{9}\right)^{\log_3 n + 1} - 1}{\frac{2}{9} - 1}$$

$$= n^2\,\frac{1 - \frac{2}{9}\left(\frac{2}{9}\right)^{\log_3 n}}{\frac{7}{9}} = n^2\,\frac{9 - 2\cdot\left(\frac{2}{9}\right)^{\log_3 n}}{7} =$$

$$= n^2 \cdot \frac{9 - 2\cdot\frac{2^{\log_3 n}}{9^{\log_3 n}}}{7} = n^2 \cdot \frac{9 - 2\cdot\frac{n^{\log_3 2}}{3^{2\log_3 n}}}{7}$$

$$= n^2\,\frac{9 - \frac{2n^{\log_3 2}}{3^{\log_3 n^2}}}{7} = n^2\,\frac{9 - \frac{2n^{\log_3 2}}{n^2}}{7} = n^2\,\frac{9n^2 - 2n^{\log_3 2}}{7n^2}$$

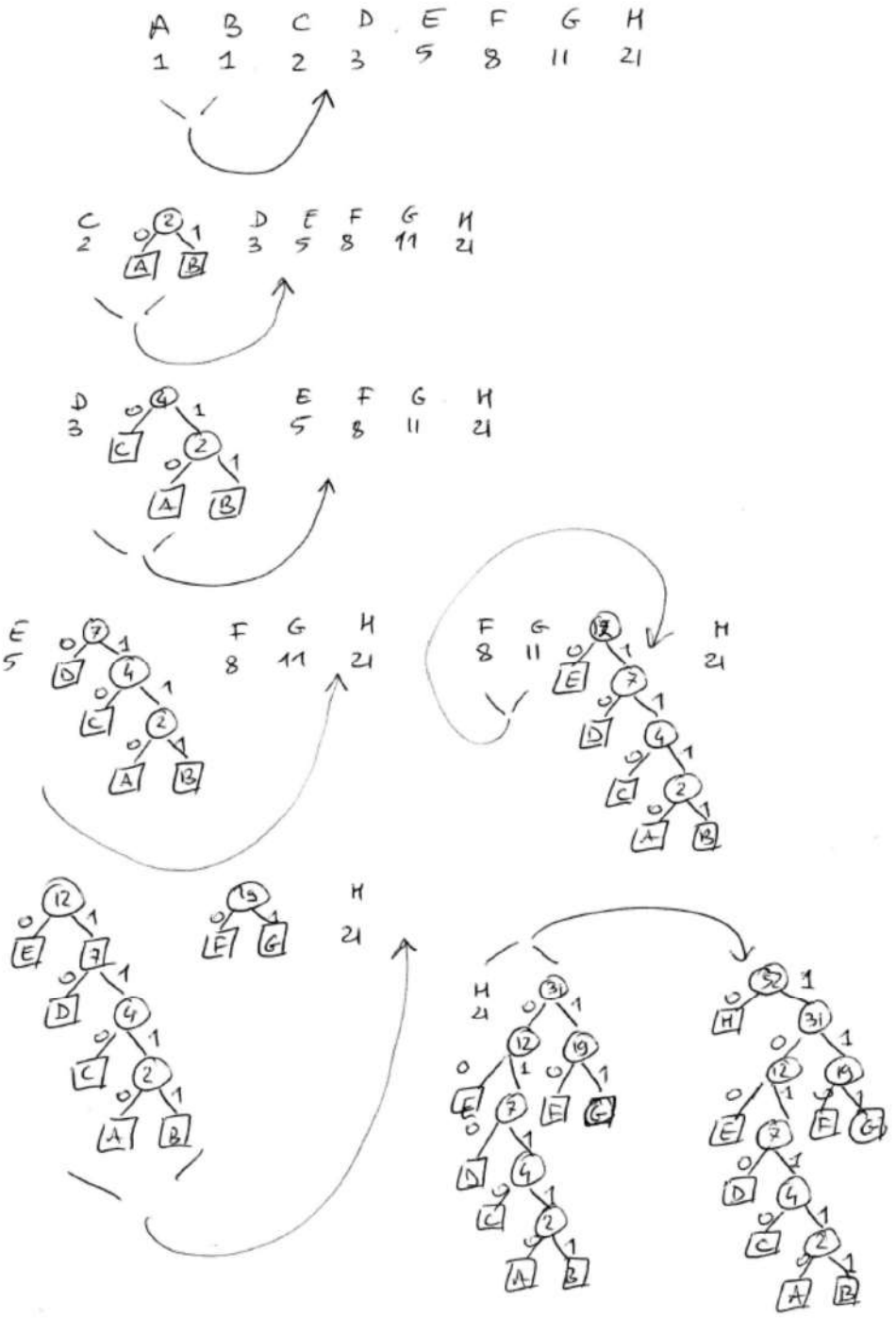
$$T(n) = \vartheta(n^2)$$

Codici di Huffman

8.2.3 Esercizio proposto pag. 122

Si determini un codice di Huffman ottimo per i seguenti caratteri con le frequenze specificate.

A:1 B:1 C:2 D:3 E:5 F:8 G:11 H:21



Alberi binari

4.1.4 Esercizio proposto pag. 47

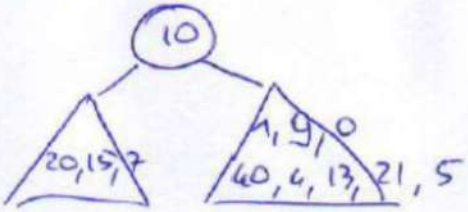
Sia dato un albero binario con 12 nodi. Nella visita si ottengono le tre seguenti sequenze di chiavi:

pre-order:	10	15	20	7	40	9	1	0	13	4	21	5
post-order:	20	7	15	1	0	9	4	5	21	13	40	10
in-order:	20	15	7	10	1	9	0	40	4	13	21	5

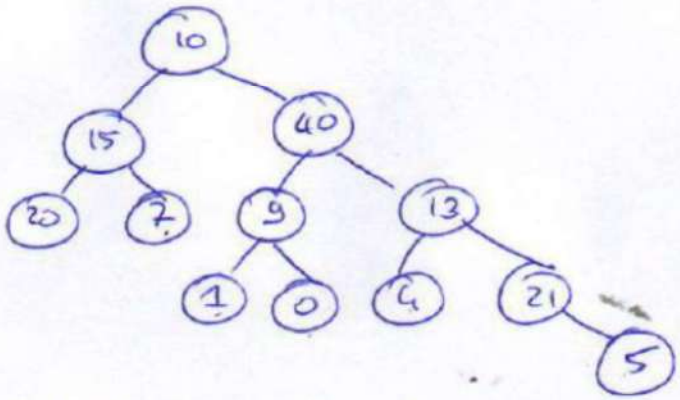
Si disegni l'albero binario di partenza.

da preorder ricavo root = 10

da inorder: a L 20, 15, 7 a R 1, 9, 0, 40, 4, 13, 21, 5



successivamente ricavo



verifico se compatibile con postorder

20 7 15 1 0 9 4 5 21 13 40 10

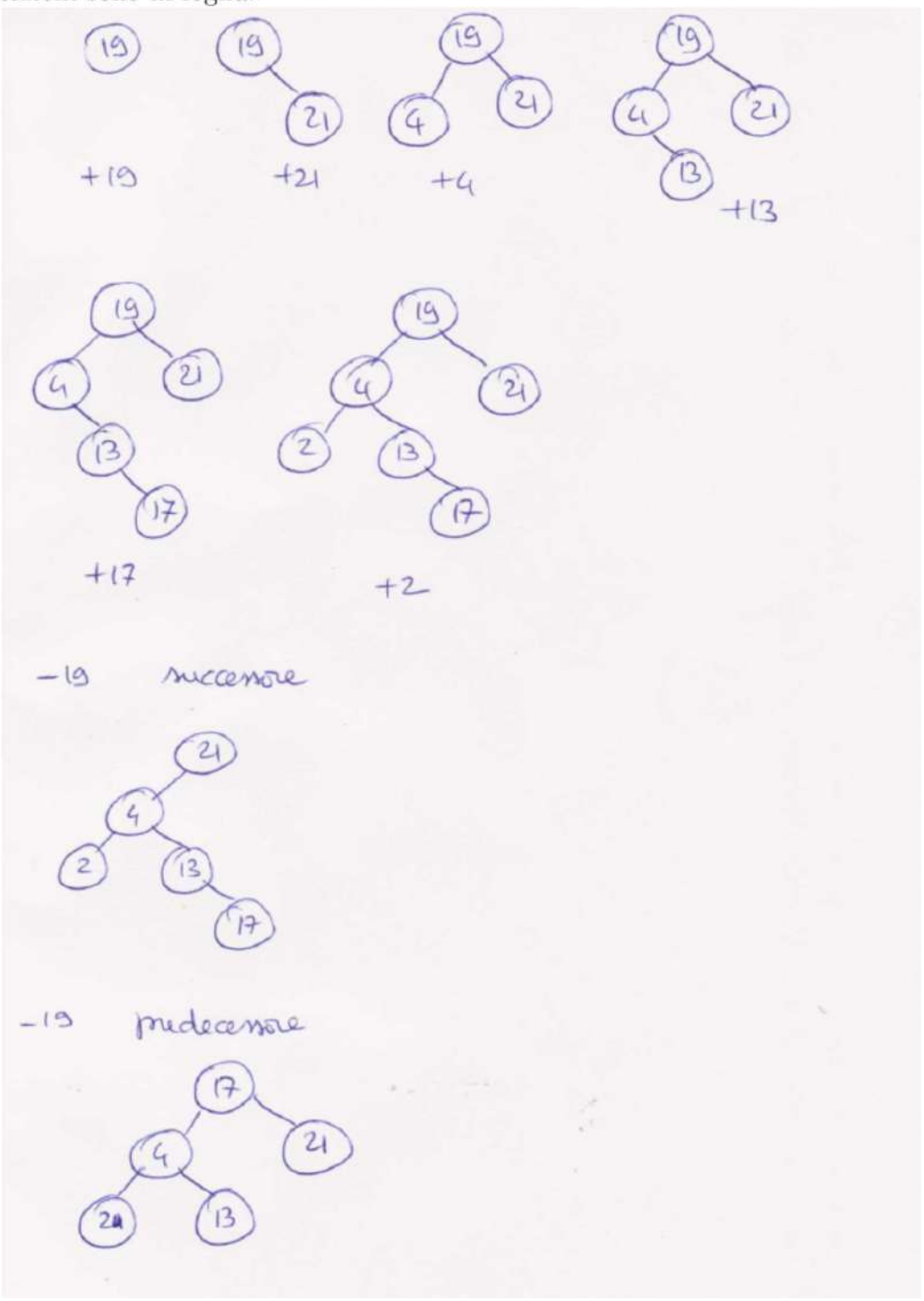
BST: inserzione in foglia/estrazione

5.2.3 Esercizio proposto pag. 70

Si effettuino secondo l'ordine specificato le seguenti operazioni di inserzione (+) ed estrazione (−) su un BST supposto inizialmente vuoto:

$$+19 \quad +21 \quad +4 \quad +13 \quad +17 \quad +2 \quad -19$$

Le inserzioni sono in foglia.



BST: inserzione in radice

5.2.3 Esercizio proposto pag. 70

Si inseriscano in radice nel BST di Figura 5.18 la sequenza le chiavi:

$+11 \quad +4 \quad +9 \quad +25$

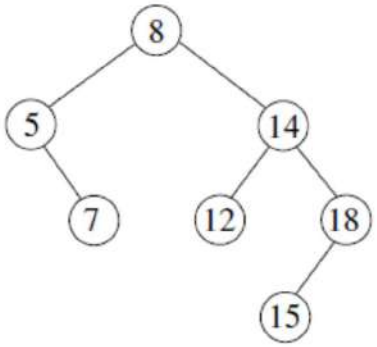
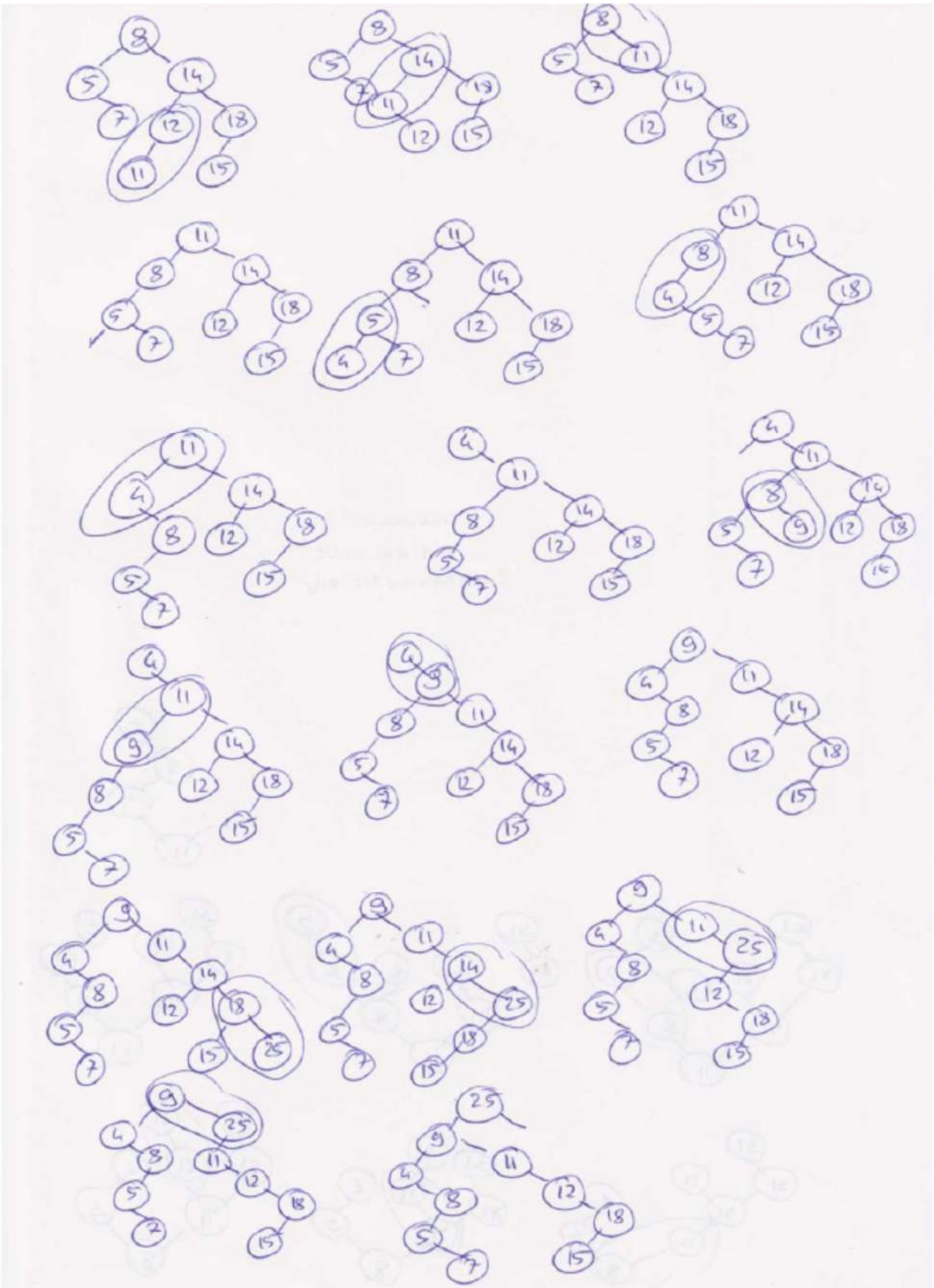


Figura 5.18



I-BST: inserzione in foglia/ricerca

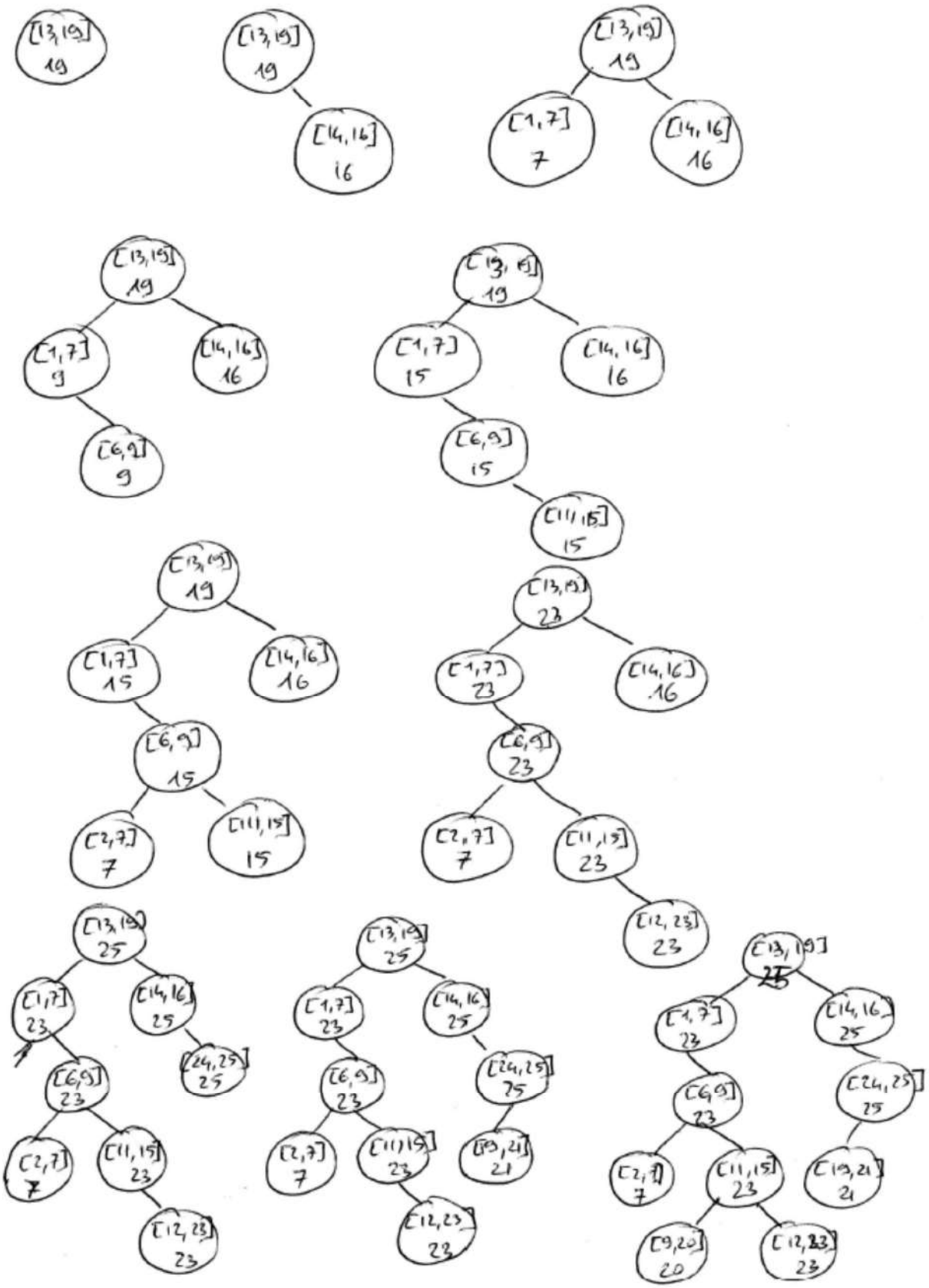
5.3.3 Esercizio proposto pag. 78

Si effettuino secondo l'ordine specificato le seguenti operazioni di inserzione su un Interval BST supposto inizialmente vuoto:

[13,19] [14,16] [1, 7] [6, 9] [11,15] [2, 7] [12,23] [24, 25] [19, 21] [9,20]

Si identifichi quindi il primo nodo con intersezione non nulla con l'intervallo [22,24].
Le inserzioni vengano effettuate nelle foglie.

[13,19] [14,16] [1,7] [6,9] [11,15] [2,7] [12,23] [24,25] [19,21] [9,20]



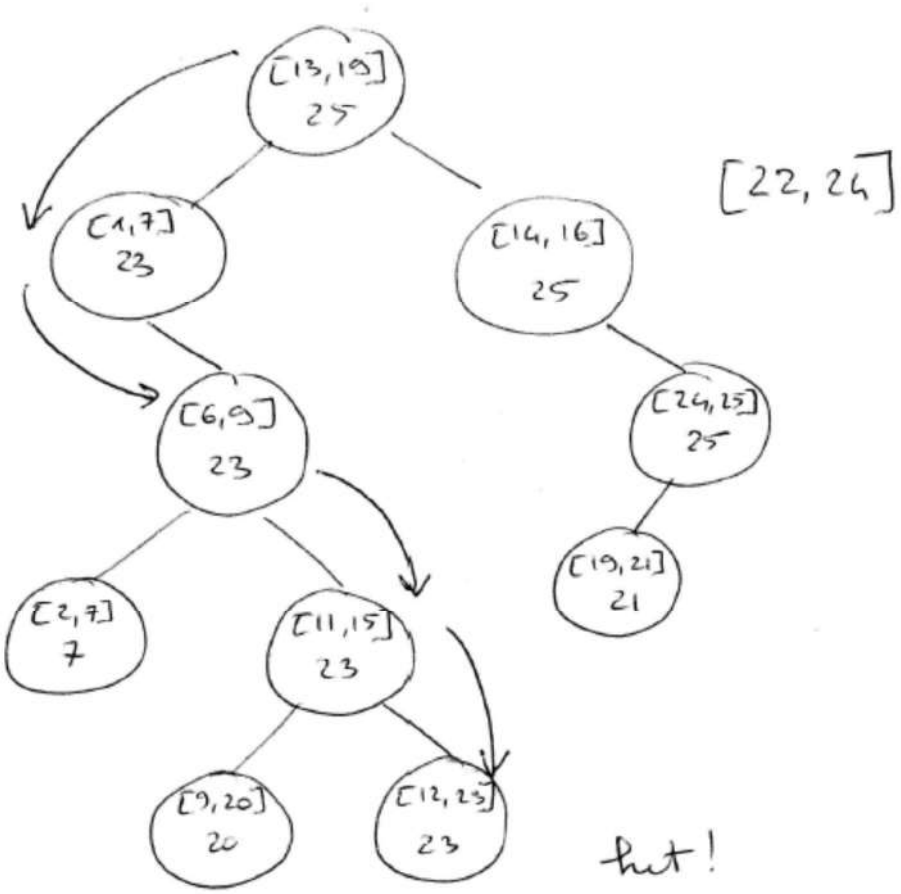


Tabelle di hash: open addressing con linear probing

6.4.4 Esercizio proposto pag. 88

Sia data la sequenza di chiavi

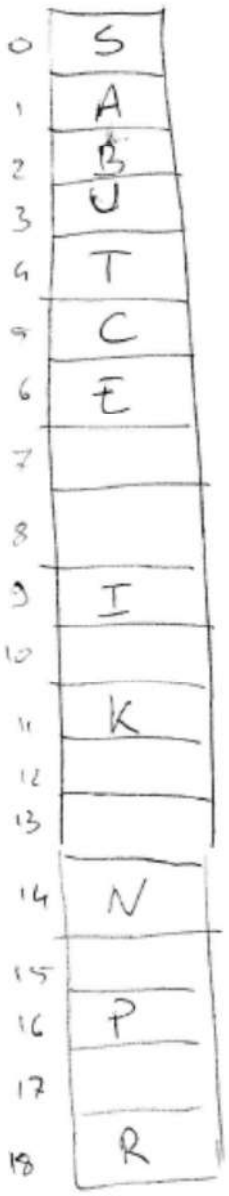
$$B \quad A \quad N \quad K \quad R \quad U \quad P \quad T \quad C \quad I \quad E \quad S$$

dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto inglese ($A = 1, \ldots, Z = 26$). Si riporti la struttura di una tabella di hash di dimensione 19, e 3

inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza di cui sopra. Si supponga di utilizzare l'open addressing con linear probing e che la funzione di hash sia $h(k) = k \ \% \ 19$.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

BANKRUPTCIES



$h(U) = 2$ collisione, inserzione in 3

$h(T) =$ 1, poi 2 e 3 con collisione " 4

$h(C) = 3$ " " " 5

$h(E) = 5$ " " " 6

Tabelle di hash: open addressing con quadratic probing

6.5.4 Esercizio proposto pag. 93

Sia data la sequenza di chiavi

$$T \quad R \quad O \quad U \quad B \quad L \quad E \quad M \quad A \quad K \quad I \quad N \quad G \quad S$$

dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto inglese ($A = 1, \ldots, Z = 26$). Si riporti la struttura di una tabella di hash di dimensione 17, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza di cui sopra. Si supponga di utilizzare l'open addressing con con quadratic probing ($c_1 = 0$ e $c_2 = 1$).

TROUBLEMAKINGS

start = h(k)

index = (start + i^2) % 17

0	
1	R
2	B
3	T
4	U
5	E
6	S
7	G
8	
9	I
10	A
11	K
12	L
13	M
14	N
15	O
16	

h(A) = 1 collision

index = (1 + 1^2) % 17 = 2

index = (1 + 2^2) % 17 = 5

index = (1 + 3^2) % 17 = 10

h(S) = 2 collision

index = (2 + 1^2) % = 3

index = (2 + 2^2) % = 6

Tabelle di hash: open addressing con double hashing

6.6.4 Esercizio proposto pag. 98

Sia data la sequenza di chiavi intere

3 16 9 22 15 28 21 34 27 40 33 46 39 52 45

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 double hashing. Sia a carico del candidato la scelta di $h_1(k)$ e $h_2(k)$.

3 16 9 22 15 28 21 34 27 40 33 46 39 52 45

$h_1(k) = k \bmod 19$

$h_2(k) = 1 + k \bmod 97$

$h(k) = (k \bmod 19 + (1 + k \bmod 97)) \bmod 19$

0	28
1	39
2	21
3	3
4	45
5	40
6	
7	22
8	27
9	9
10	52
11	
12	34
13	~~33~~
14	33
15	15
16	16
17	46
18	

$h(22) = 3$ collisione

$i = (22 \bmod 19 + 1 + 22 \bmod 97) \bmod 19 = 7$

$h(28) = 9$ collisione

$i = (28 \bmod 19 + 1 + 28 \bmod 97) \bmod 19 = 0$

$h(34) = 15$ collisione

$i = (34 \bmod 19 + 1 + 34 \bmod 97) \bmod 19 = 12$

$h(40) = 2$ collisione

$i = (40 \bmod 19 + 1 + 40 \bmod 97) \bmod 19 = 5$

$h(46) = 8$ collisione

$i = (46 \bmod 19 + 1 + 46 \bmod 97) \bmod 19 = 17$

$h(52) = 14$ collisione

$i = (52 \bmod 19 + 1 + 52 \bmod 97) \bmod 19 = 10$

$h(45) = 7$ collisione

$i = (45 \bmod 19 + 1 + 45 \bmod 97) \bmod 19 = 15$ collisione

$i = (15 + 1 + 45 \bmod 97) \bmod 19 = 4$

Heapsort

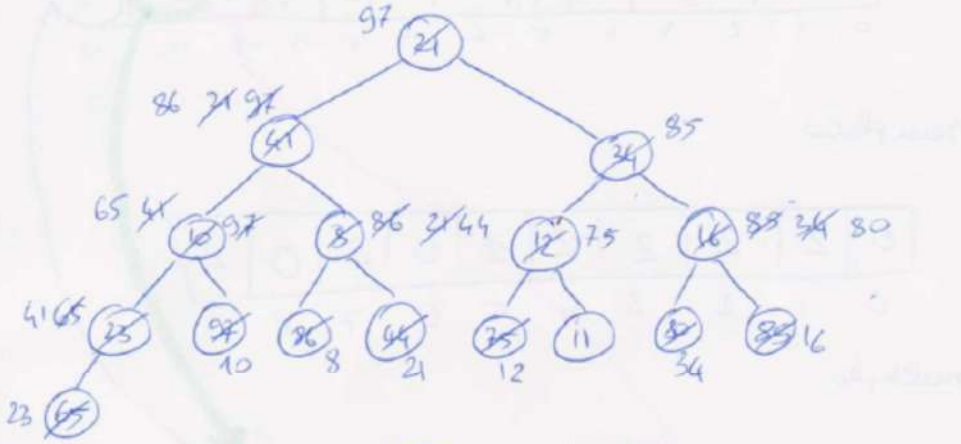
7.2.4 Esercizio proposto pag. 104

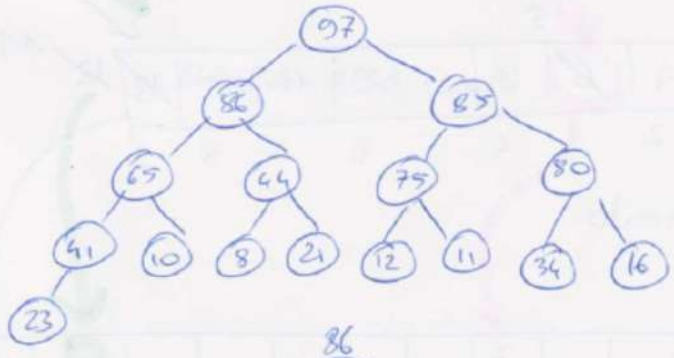
Sia data la sequenza di interi, supposta memorizzata in un vettore:

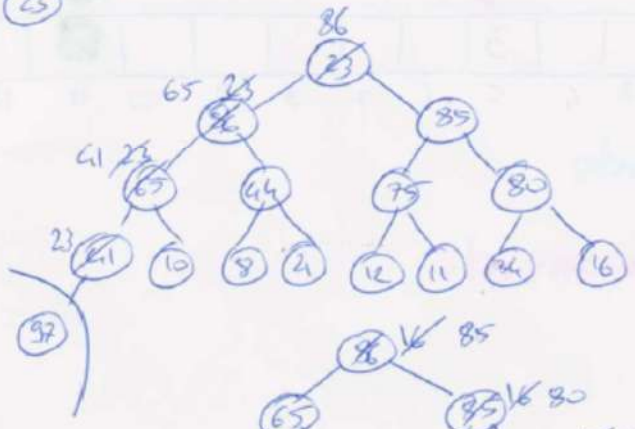
21 41 34 10 8 12 16 23 97 86 44 75 11 80 85 65

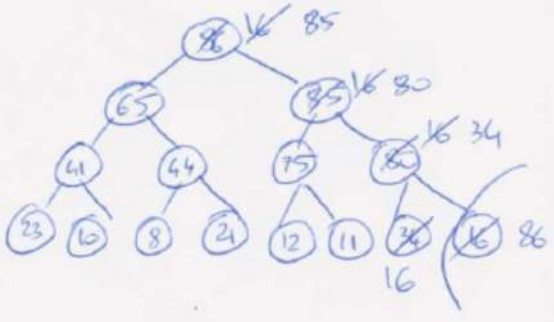
- la si trasformi in uno heap, ipotizzando di usare un vettore come struttura dati. Si riportino graficamente i diversi passi della costruzione dello heap e il risultato finale. Si ipotizzi che nella radice dello heap sia memorizzato il valore massimo
- si eseguano su tale heap i primi due passi dell'algoritmo di heap sort. Si osservi che la sequenza è già memorizzata nel vettore e rappresenta una configurazione intermedia per cui la proprietà di heap non è ancora soddisfatta.

21	41	34	10	8	12	16	23	97	86	44	75	11	80	85	65









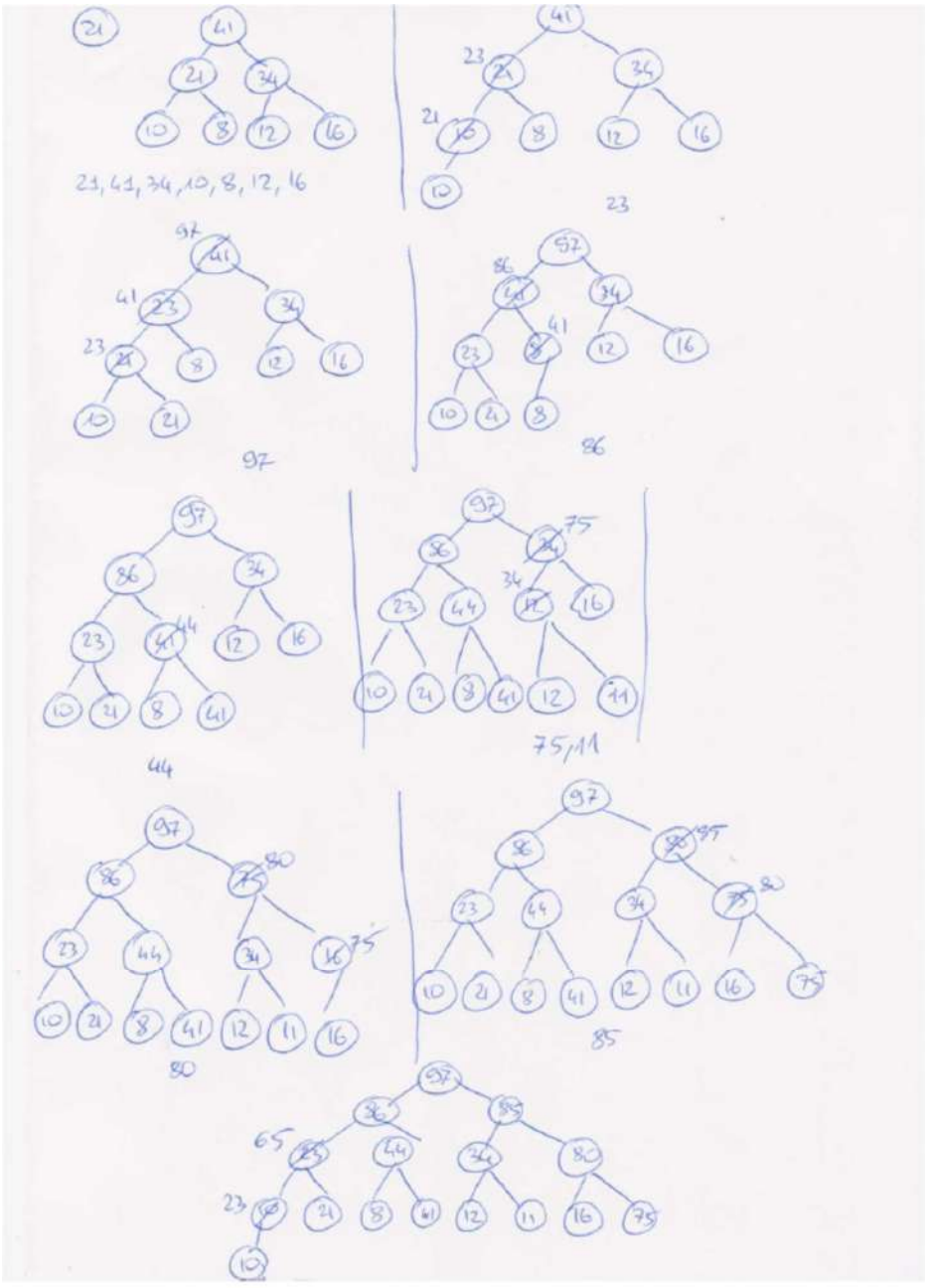
Code a priorità di dati

7.3.4 Esercizio proposto pag. 116

Si inserisca la seguente sequenza di interi in una coda a priorità di dati inizialmente supposta vuota:

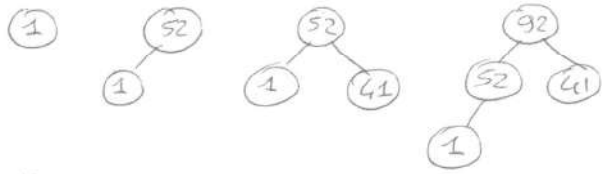
21 41 34 10 8 12 16 23 97 86 44 75 11 80 85 65

Si ipotizzi di usare uno heap come struttura dati. Si disegni la struttura ai diversi passi dell'inserzione. Si ipotizzi che la priorità massima sia associata alla chiave a valore massimo.

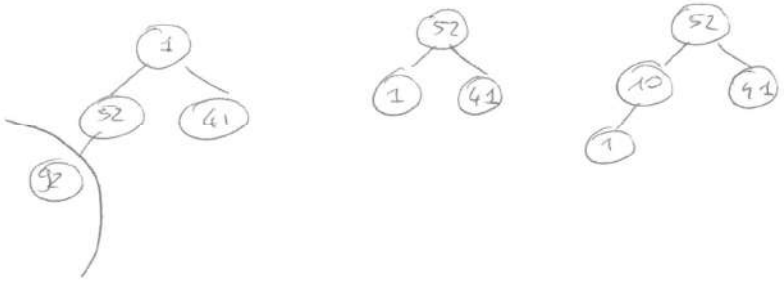


2 (2 punti)

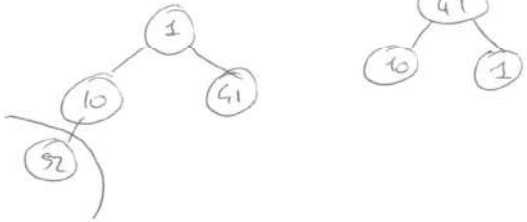
Sia data una coda a priorità inizialmente vuota implementata mediante uno heap. Sia data la sequenza di interi e carattere *: 1 52 41 92 * 10 * * 4 9 * 93 * * * 31 78 29 dove ad ogni intero corrisponde un inserimento nella coda a priorità e al carattere * un'estrazione con cancellazione del massimo. Si riporti la configurazione della coda a priorità dopo ogni operazione e la sequenza dei valori restituiti dalle estrazioni con cancellazione del massimo.



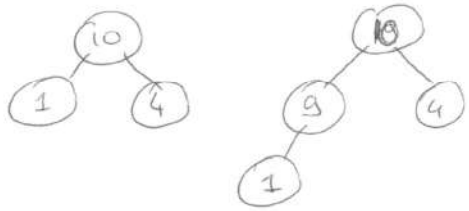
cancello il max



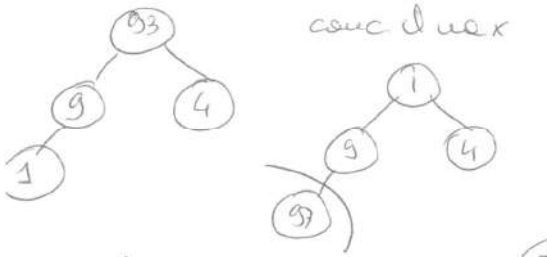
cancello il max



cancello il max

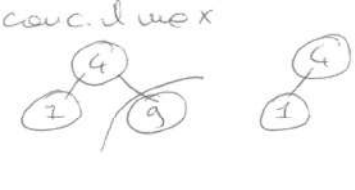


cancello il max

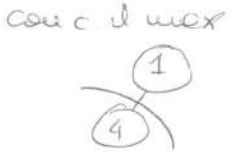


cancello il max

cancello il max



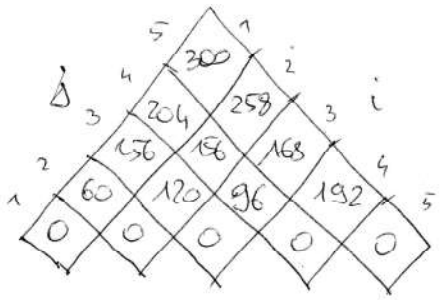
cancello il max

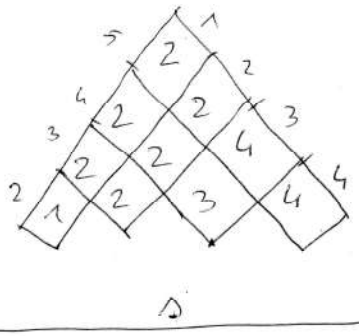


Programmazione dinamica: parentesizzazione ottima

Data la catena di matrici (A_1, A_2, A_3, A_4, A_5) di dimensioni (4x5), (5x3), (3x8), (8x4) e (4x6) rispettivamente, si determini mediante un algoritmo di programmazione dinamica la parentesizzazione ottima del prodotto di matrici che minimizza il numero di moltiplicazioni.

p_0	4
p_1	5
p_2	3
p_3	8
p_4	4
p_5	6





$m_{12} = m_{11} + m_{22} + p_0 p_1 p_2 = 0+0+60 = 60$ $k=1$

$m_{23} = m_{22} + m_{33} + p_1 p_2 p_3 = 0+0+120 = 120$ $k=2$

$m_{34} = m_{33} + m_{44} + p_2 p_3 p_4 = 0+0+96 = 96$ $k=3$

$m_{45} = m_{44} + m_{55} + p_3 p_4 p_5 = 0+0+192 = 192$ $k=4$

m_{13}

$k=1 = m_{11} + m_{23} + p_0 p_1 p_3 = 0+120+160 = 280$

$k=2 = m_{12} + m_{33} + p_0 p_2 p_3 = 60+0+96 = 156$ $k=2$

m_{24}

$k=2 = m_{22} + m_{34} + p_1 p_2 p_4 = 0+96+60 = 156$ $k=2$

$k=3 = m_{23} + m_{44} + p_1 p_3 p_4 = 120+0+160 = 280$

m_{35}

$k=3 = m_{33} + m_{45} + p_2 p_3 p_5 = 0+192+144 = 336$

$k=4 = m_{34} + m_{55} + p_2 p_4 p_5 = 96+0+72 = 168$ $k=4$

m_{14}

$k=1 = m_{11} + m_{24} + p_0 p_1 p_4 = 0+156+80 = 236$

$k=2 = m_{12} + m_{34} + p_0 p_2 p_4 = 60+96+48 = 204$ $k=2$

$k=3 = m_{13} + m_{44} + p_0 p_3 p_4 = 156+0+128 = 284$

m_{25}

$k=2 = m_{22} + m_{35} + p_1 p_3 p_5 = 0+168+90 = 258$

$k=3 = m_{23} + m_{45} + p_1 p_3 p_5 = 120+192+240 = 552$ $k=2$

$k=4 = m_{24} + m_{55} + p_1 p_4 p_5 = 156+0+240 = 396$

m_{15}

$= m_{11} + m_{25} + p_0 p_2 p_5 =$ 0 +258 +120 = 378

$k=1$

$k=2 \quad = m_{12} + m_{35} + p_0 p_2 p_5 = 60 + 168 + 72 = 300$

$k=3 \quad = m_{13} + m_{45} + p_0 p_3 p_5 = 156 + 192 + 192 = 540$

$k=4 \quad = m_{14} + m_{55} + p_0 p_4 p_5 = 204 + 0 + 96 = 300$

$k=2$ o $k=4$

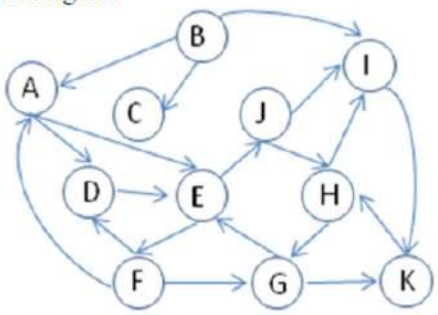
scelta k = 2

parentesizzazione ottima ((A1 x A2) x ((A3xA4)x A5)

Visite di grafi

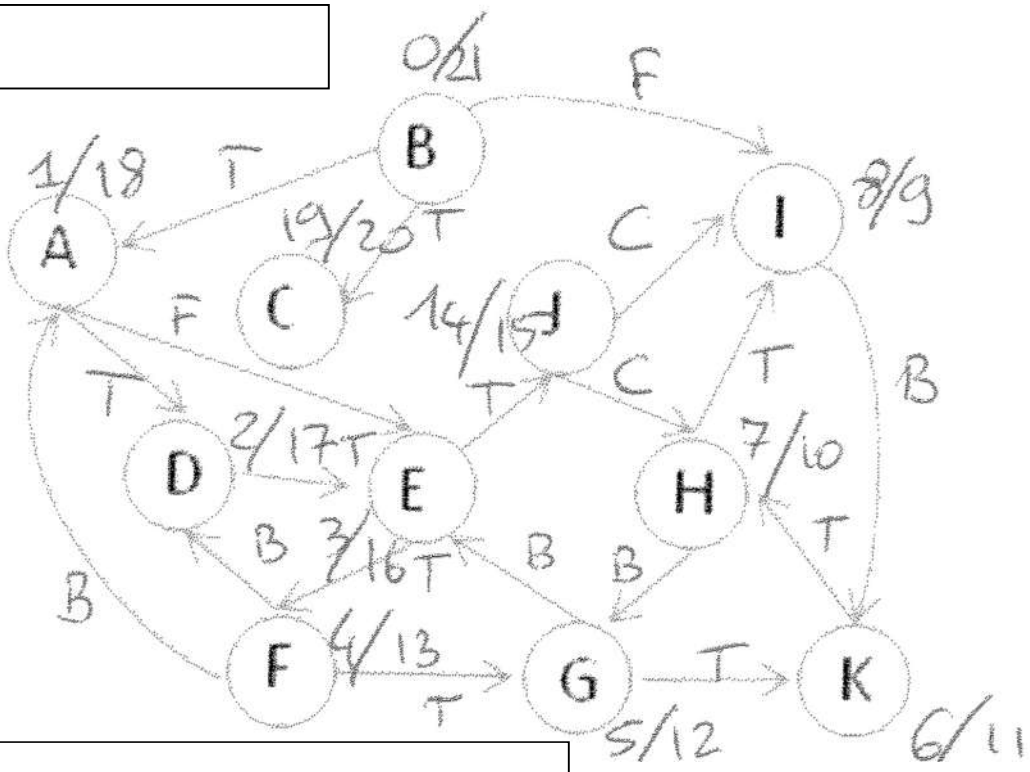
5

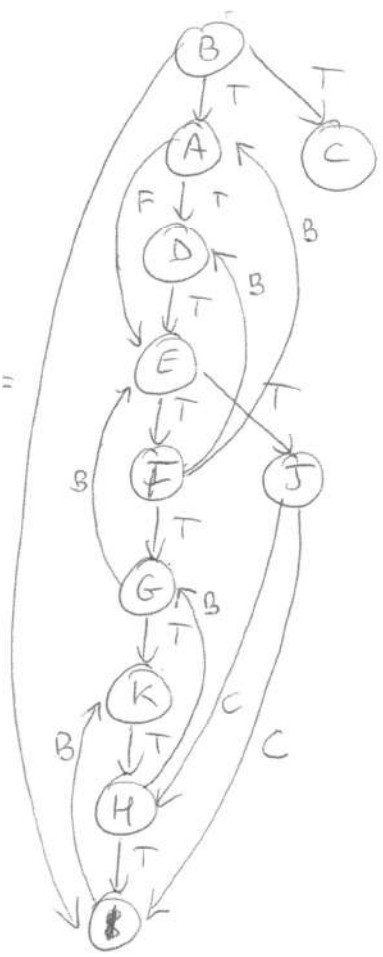
Sia dato il seguente grafo orientato in figura:

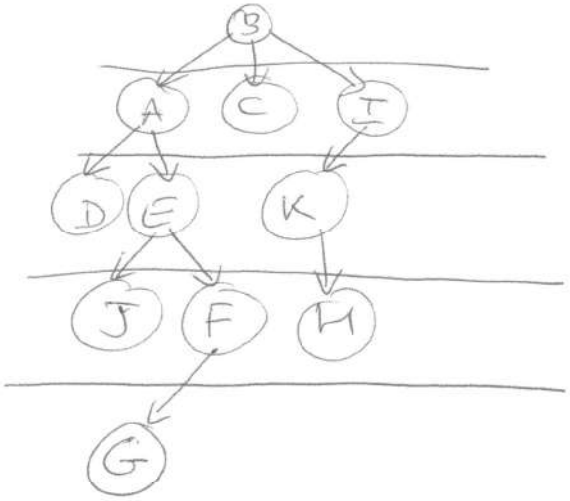


- se ne effettui una visita in profondità, considerando **B** come vertice di partenza. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico. Si etichettino indicando per ognuno di essi i tempi di scoperta e di fine elaborazione nel formato tempo1/tempo2 (**2 punti**)
- se ne effettui una visita in ampiezza, considerando **B** come vertice di partenza. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico (**1.5 punti**)
- lo si ridisegni, etichettando ogni suo arco come T (tree), B (back), F (forward), C (cross), considerando **B** come vertice di partenza (**1.5 punti**).

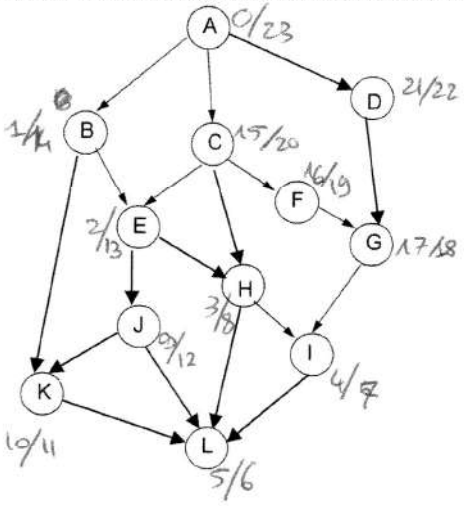
Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

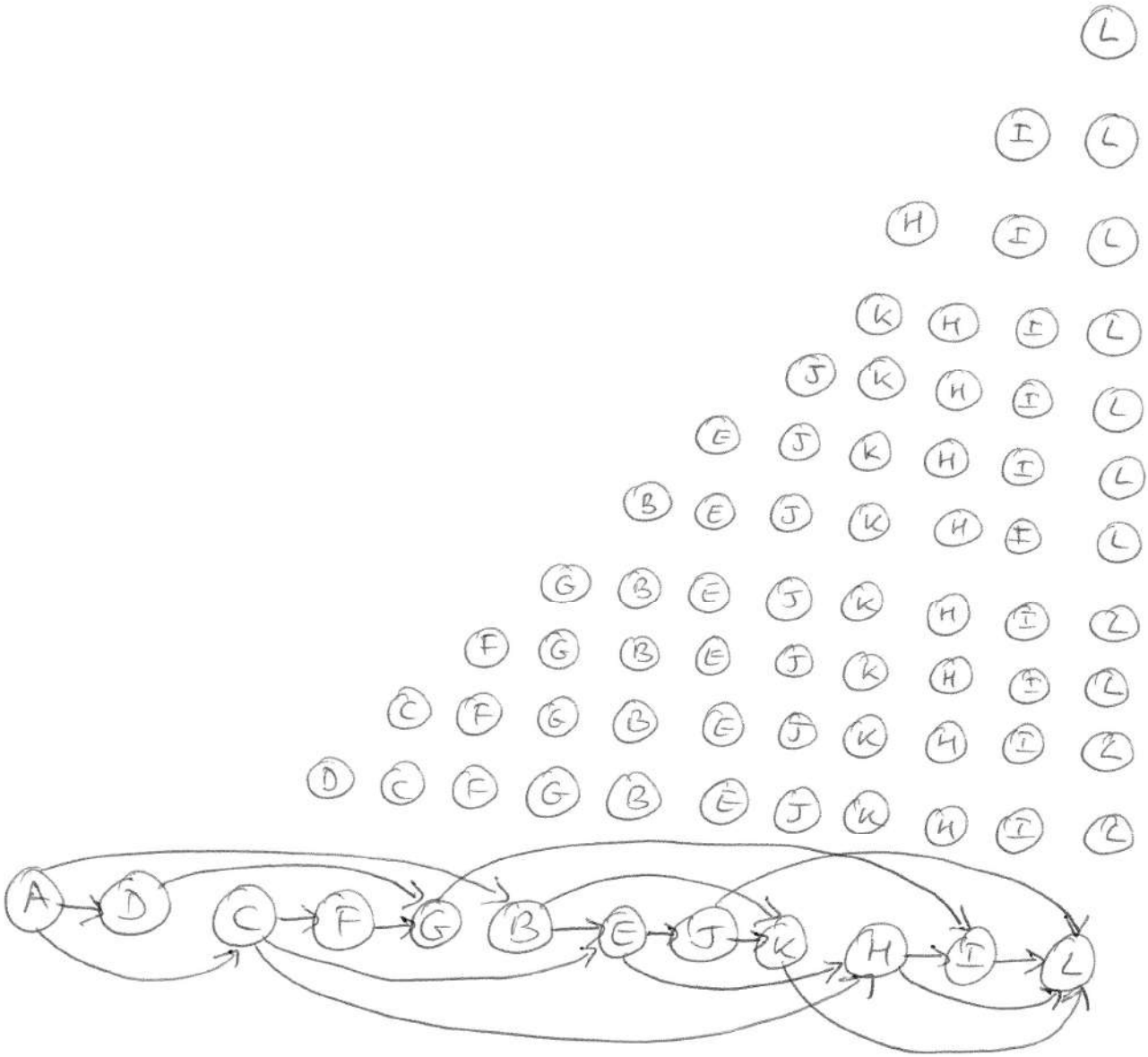

0/21
F
B
1/18
T
A
19/20T
C
I
8/9
C
F
C
14/15J
T
T
C
T
B
T
D
2/17T
E
H
7/10
T
B
3/16T
B
B
B
F
4/13
T
G
T
K
5/12
6/11


B
A
C
D
E
F
J
G
K
H
T
F
B
C


B
A
C
T
D
E
K
J
F
H
G

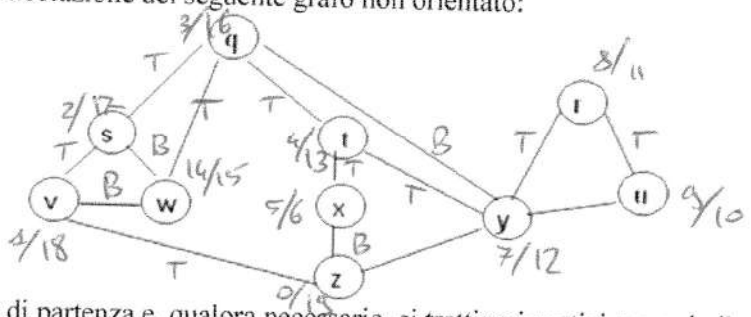
Si ordini topologicamente il seguente DAG. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico e si assuma che la lista delle adiacenze sia anch'essa ordinata alfabeticamente.



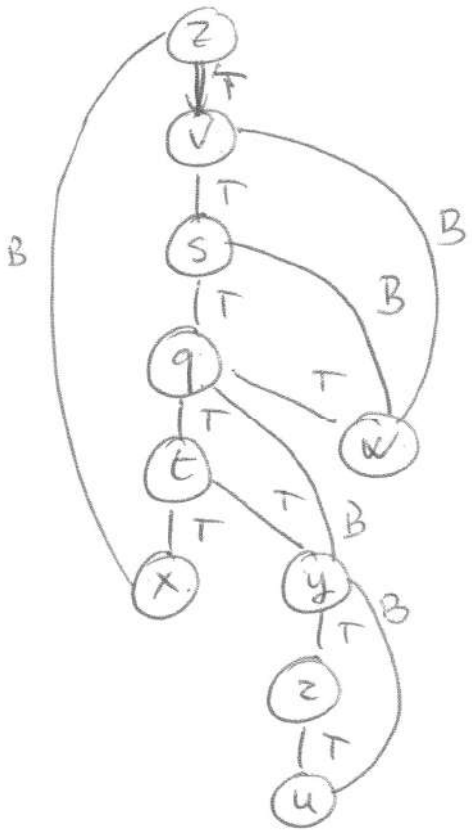


5 (2 punti)

Si determinino i punti di articolazione del seguente grafo non orientato:



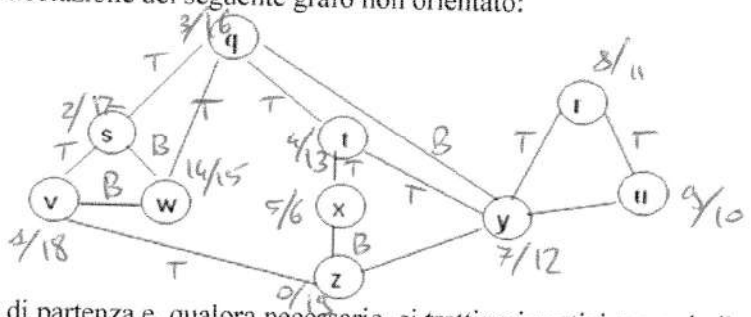
Si consideri **z** come vertice di partenza e, qualora necessario, si trattino i vertici secondo l'ordine alfabetico e si assuma che la lista delle adiacenze sia anch'essa ordinata alfabeticamente.



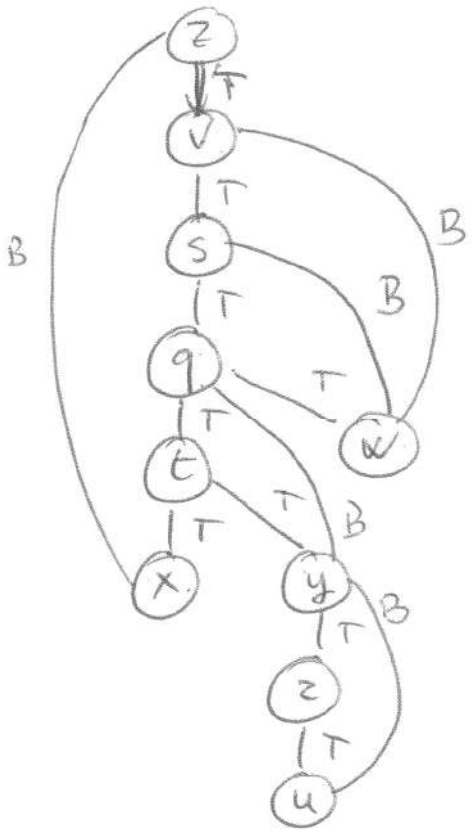
y è nodo etc.

5 (2 punti)

Si determinino i punti di articolazione del seguente grafo non orientato:



Si consideri **z** come vertice di partenza e, qualora necessario, si trattino i vertici secondo l'ordine alfabetico e si assuma che la lista delle adiacenze sia anch'essa ordinata alfabeticamente.

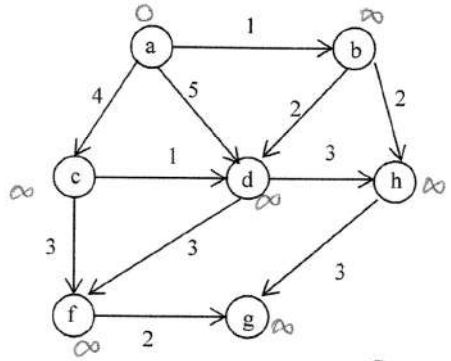


y è nodo etc.

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.

$S = \{\}$

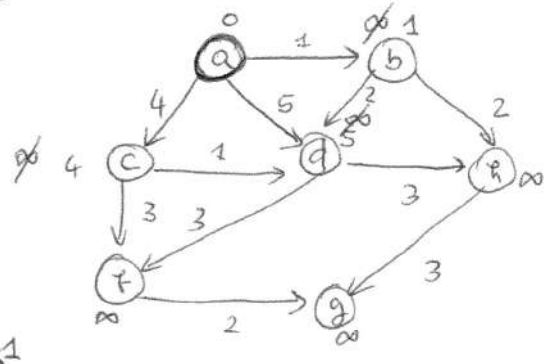
$PQ = \{a/0, b/\infty, c/\infty, d/\infty, e/\infty, g/\infty, h/\infty\}$



relax (a,b), (a,c), (a,d)

$S = \{a\}$

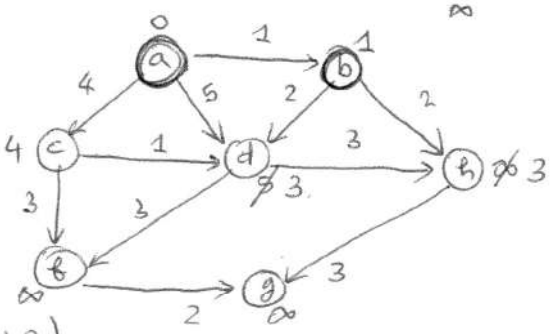
$PQ = \{b/1, c/4, d/5, e/\infty, g/\infty, h/\infty\}$



relax (b,d), (b,h)

$S = \{a, b\}$ PQ

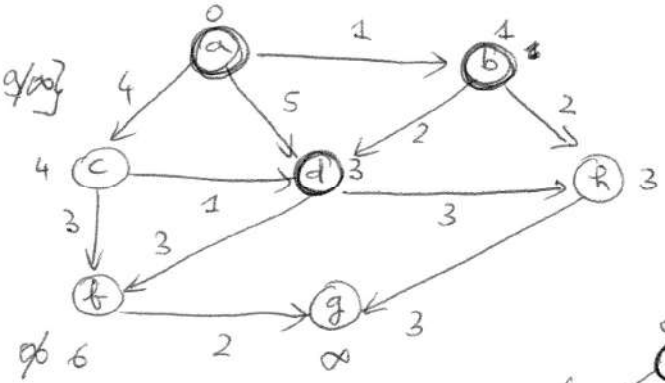
$PQ = \{d/3, h/3, c/4, e/\infty, g/\infty\}$



relax (d,e) (d,h)

$S = \{a, b, d\}$

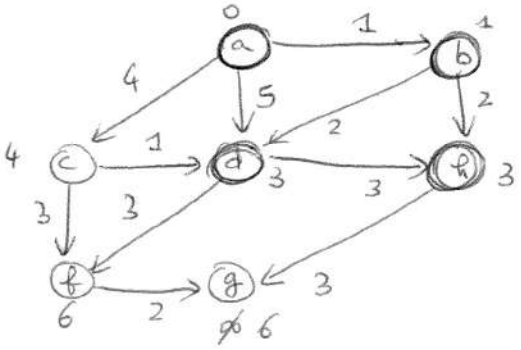
$PQ = \{h/3, c/4, e/6, g/\infty\}$



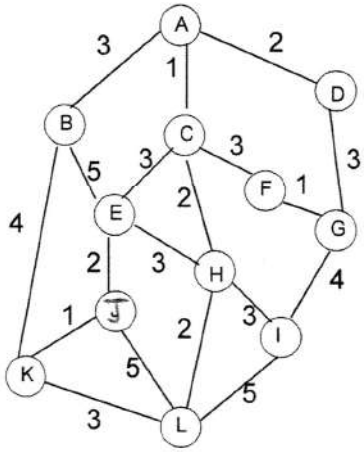
relax (h,g)

$S = \{a, b, d, h\}$

$PQ = \{c/4, e/6, g/6\}$



Sia dato il seguente grafo non orientato pesato:



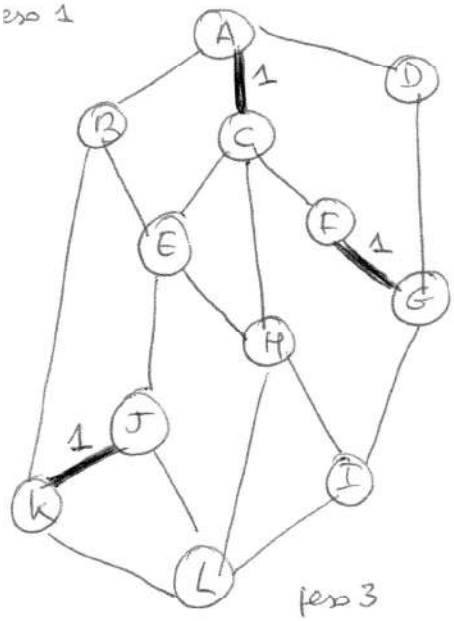
se ne determini un minimum spanning tree applicando l'algoritmo di

- Prim a partire dal vertice K
- Kruskal

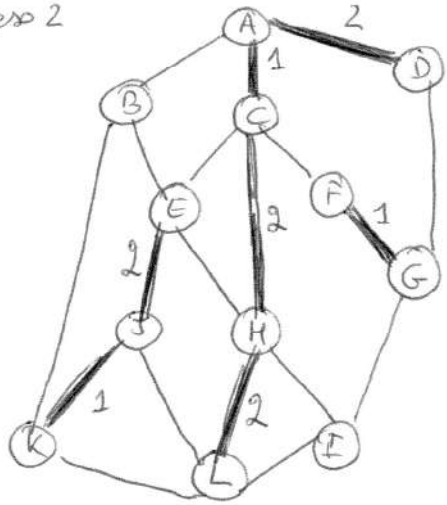
ritornando come risultato l'albero ed il valore del peso minimo. Si riportino i passi.

Kruskal

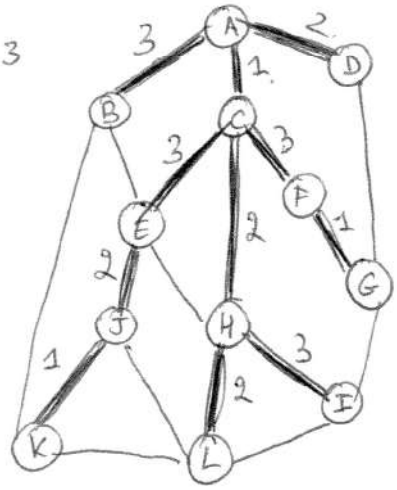
peso 1



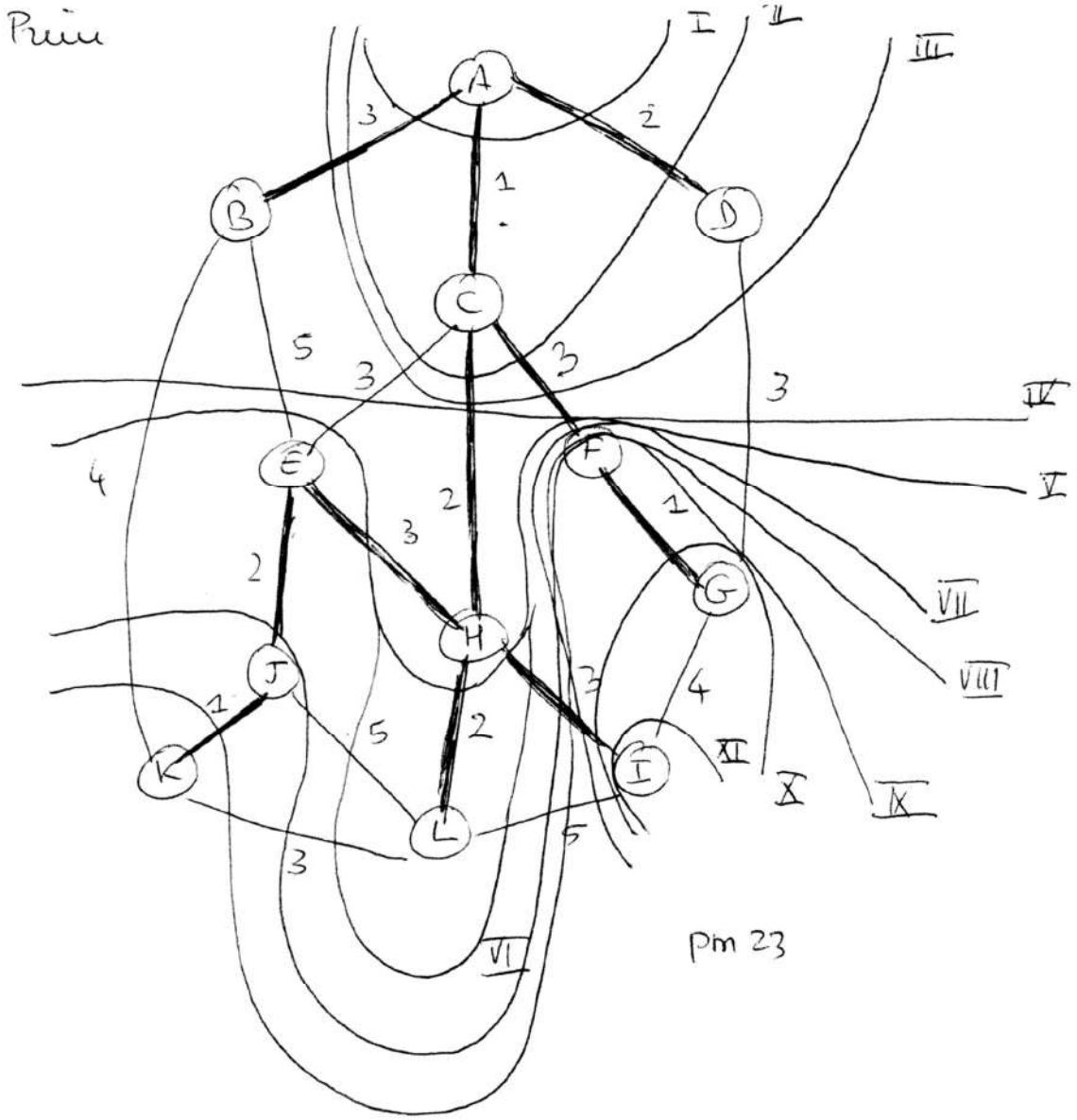
peso 2



peso 3

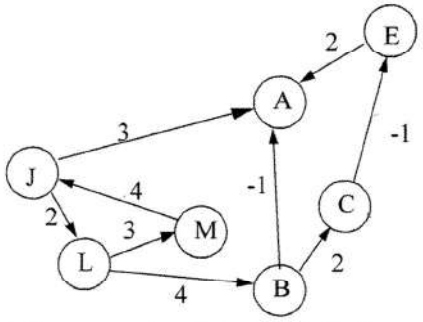


7 altre scelte che portano ad alberi diversi, ma sullo stesso peso minimo 23


Prim
A
B
C
D
E
F
G
H
I
J
K
L
I
II
III
IV
V
VI
VII
VIII
IX
X
XI
XII
pm 23

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

B-A
B-C
C-E
E-A
J-A
J-L
L-B
L-M
M-J



	0	1	2	3	4	5	6	7
A	∞	3	3	3				
B	∞	6	6	6				
C	∞	∞	8	8				
E	∞	∞	7	7				
J	⊗	0	0	0				
L	∞	2	2	2				
M	∞	5	5	5				

raggiunto il punto fisso

Si ordini topologicamente il seguente DAG. Qualora necessario, si trattino i vertici secondo l'ordine alfabetico e si assuma che la lista delle adiacenze sia anch'essa ordinata alfabeticamente.