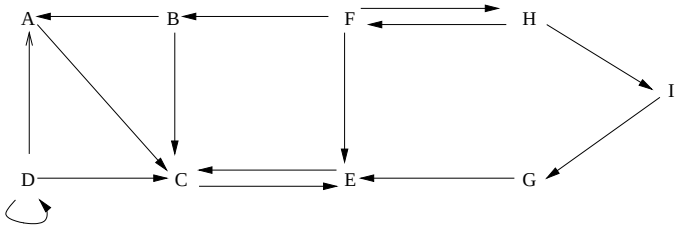
Esercizio 1)
Ordinare la seguente sequenza di chiavi applicando l'algoritmo Mergesort ed illustrando tutti i passi di esecuzione dell'algoritmo.

1	22	15	20	25	10	11	8	3	33	9	18	40	5

Esercizio 2)
Costruire l'albero di ricerca AVL bilanciato risultante dalle seguenti operazioni.
Inserimento in sequenza dei valori: `6, 15, 12, 21, 17, 25, 27, 30`.
Cancellazione in sequenza dei valori: `6, 15.`
Mostrando l'albero risultante a seguito di ogni operazione ed eseguendo eventuali operazioni di rotazione necessarie per mantenere il bilanciamento.

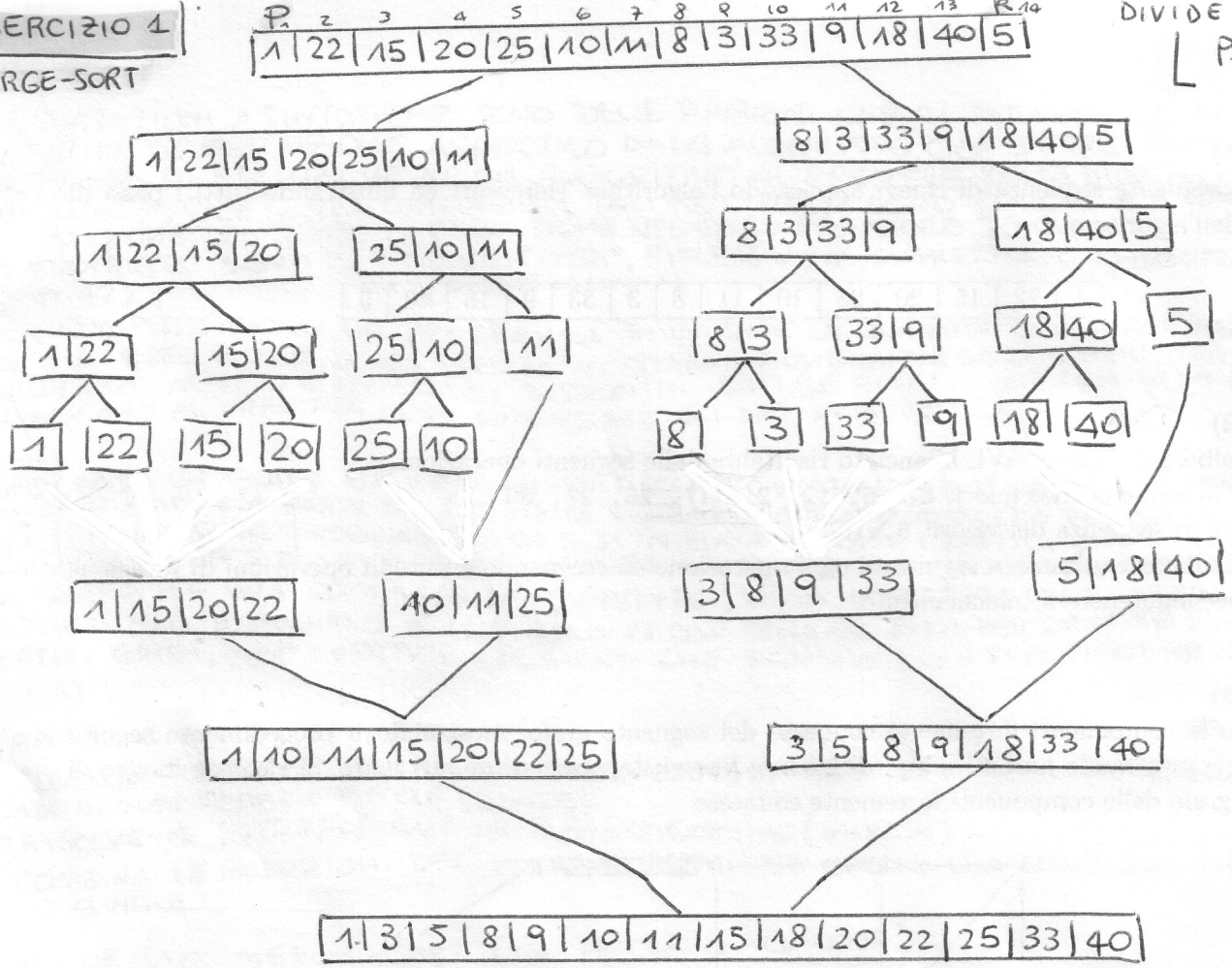
Esercizio 3)
Determinare le componenti fortemente connesse del seguente grafo, mostrando il procedimento seguito per trovarle e mostrando anche i tempi di inizio e fine visita associati ai vari nodi. Si richiede inoltre di mostrare il grafo delle componenti fortemente connesse.



ESERCIZIO 1

MERGE-SORT

DIVIDE L'ARRAY: $\left\lfloor \frac{p+r}{2} \right\rfloor$



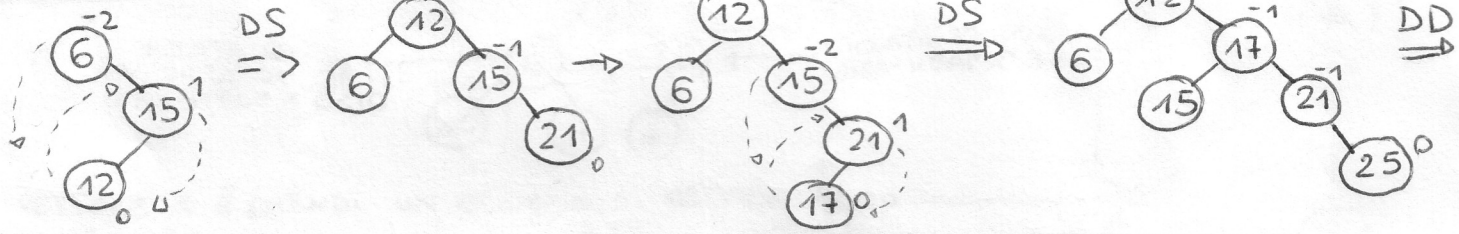
ALBERO DELLE CHIAMATE RICORSIVE

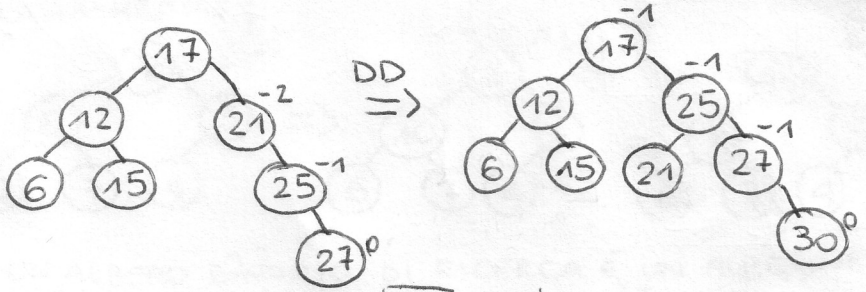
ESERCIZIO 2

INSERIMENTO: 6, 15, 12, 21, 17, 25, 27, 30

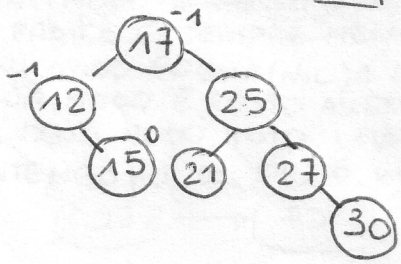
CANCELLAZIONE: 6, 15

ALBERO AVL

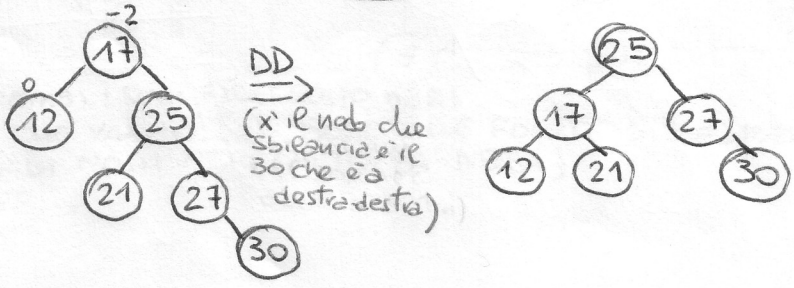


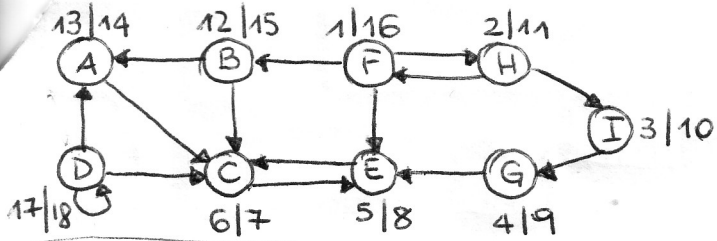


- CANCELLAZIONE: 6



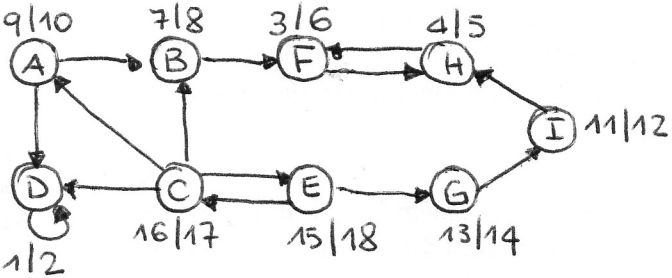
- CANCELLAZIONE: 15





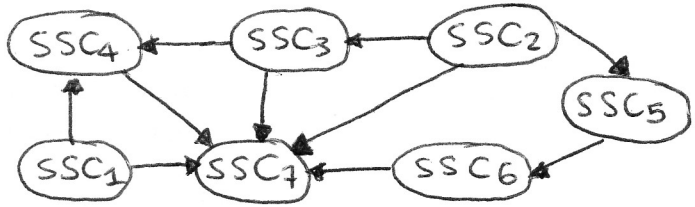
- parto dal nodo F

- GRAFO TRASPOSTO (G^T)
(si parte poi dal nodo con tempo di fine visita maggiore, e così via)



$SSC_1 = \{D\}$
$SSC_2 = \{F, H\}$
$SSC_3 = \{B\}$
$SSC_4 = \{A\}$
$SSC_5 = \{I\}$
$SSC_6 = \{G\}$
$SSC_7 = \{C, E\}$

GRAFO COMPON. FORTEM. CONNESSE:



ESERCIZIO 3

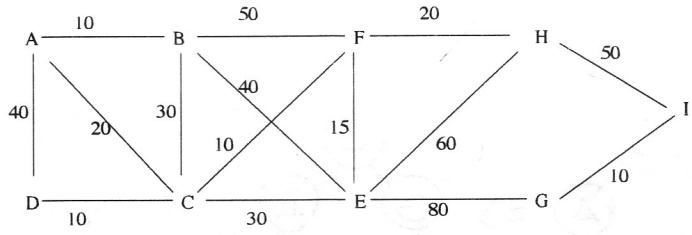
CFC

Esercizio 1) OK

Ordinare la sequenza D, A, C, B, R, I, L, P, G, F applicando l'algoritmo Heapsort. Per ogni passo di esecuzione mostrare lo heap che si ottiene dopo aver ripristinato la proprietà di max-heap.

Esercizio 2) OK

Trovare un MST per il grafo sottostante eseguengo l'algoritmo di Kruskal mostrando l'albero risultante ad ogni passo.

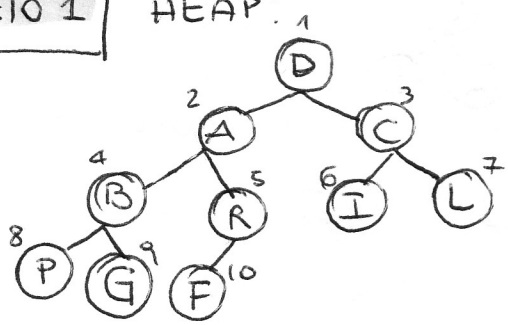


Esercizio 3)

1. Scrivere (pseudo-codice) la procedura di inserimento di un valore in una lista linkata ordinata di valori.
2. Scrivere (pseudo-codice) la procedura di ritrovamento del valore minimo in un albero binario di ricerca.

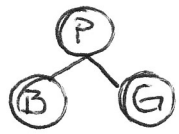
ESERCIZIO 1

HEAP:

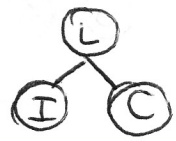


• BUILD MAX-HEAP

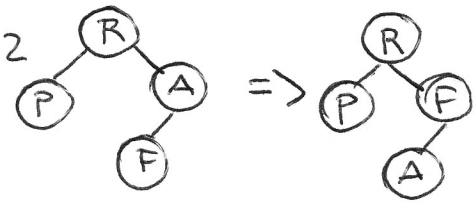
$i = 5$ niente $i = 4$



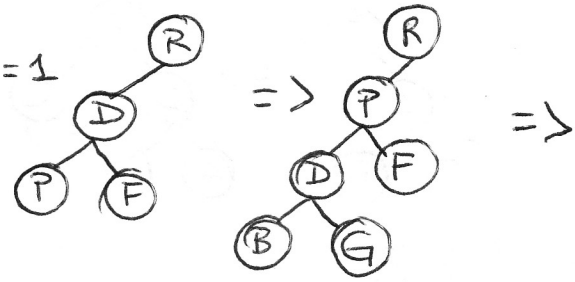
$i = 3$

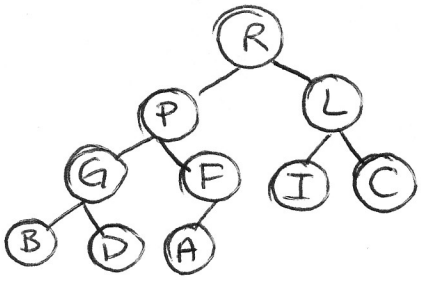


$i = 2$



$i = 1$

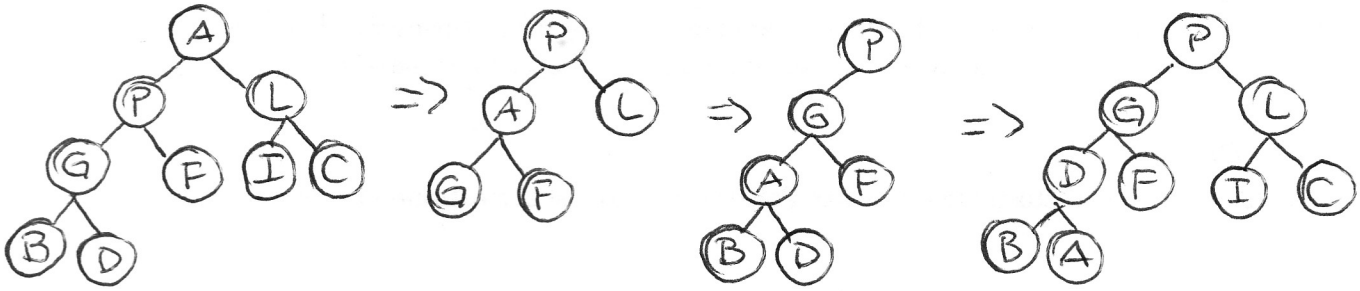




• HEAP

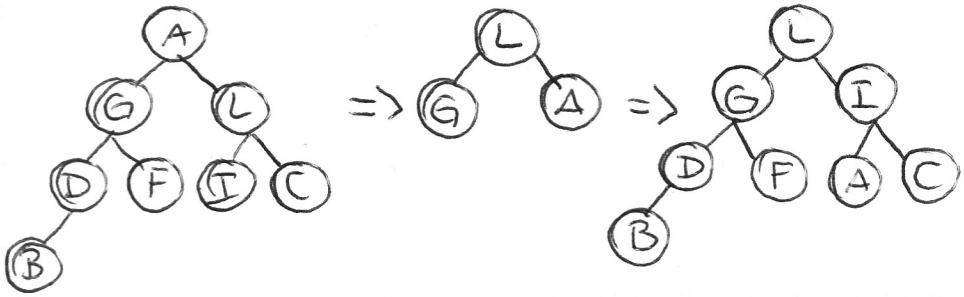
$i = 10$

10
R



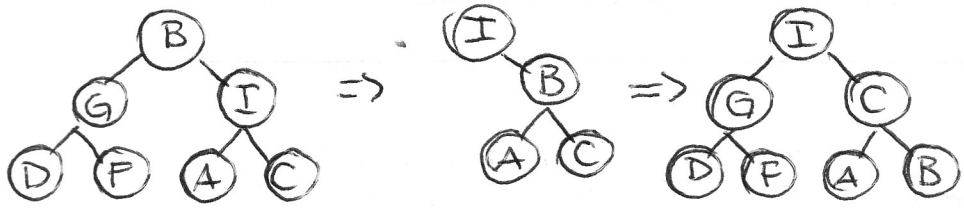
$i = 9$

9	10
P	R



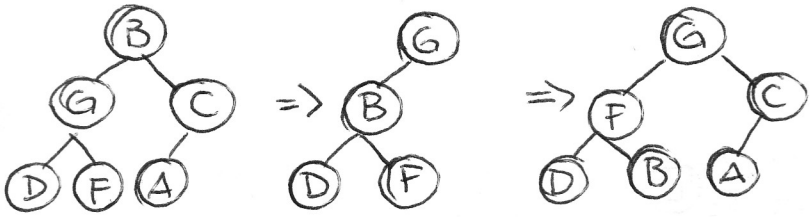
$i = 8$

8	9	10
L	P	R



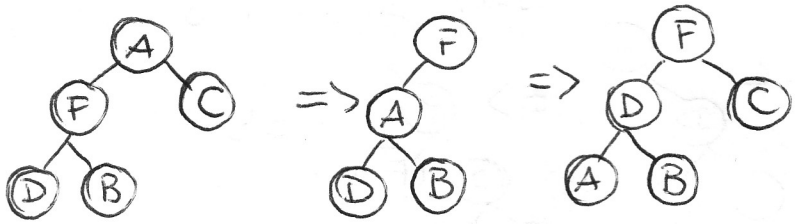
$i = 7$

7	8	9	10
I	L	P	R



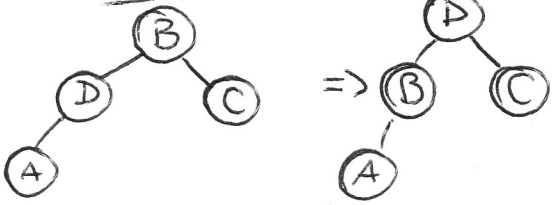
$i = 6$

6	7	8	9	10
G	I	L	P	R



$i = 5$

5	6	7	8	9	10
F	G	I	L	P	R

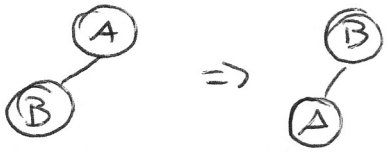


$i=4$

4	5	6	7	8	9	10
D	F	G	I	L	P	R

$i=3$

3	4	5	6	7	8	9	10
C	D	F	G	I	L	P	R



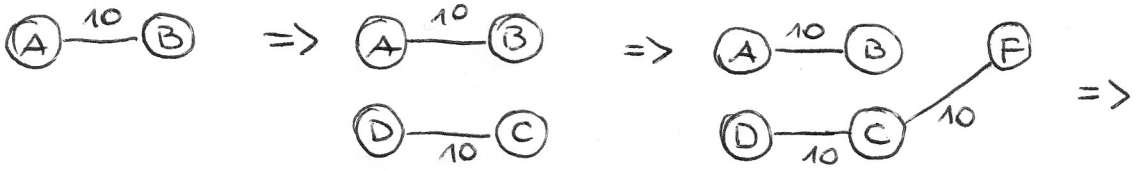
$i=2$

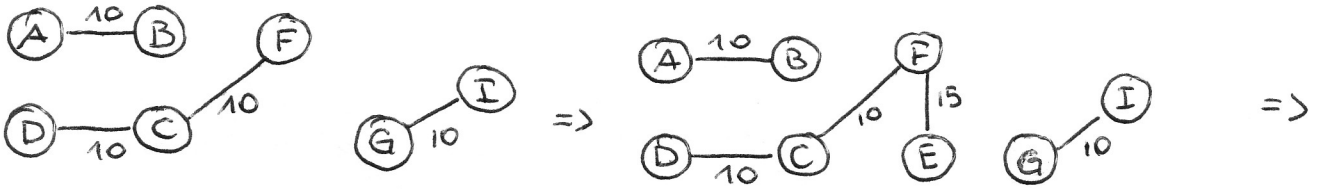
2	3	4	5	6	7	8	9	10
B	C	D	F	G	I	L	P	R

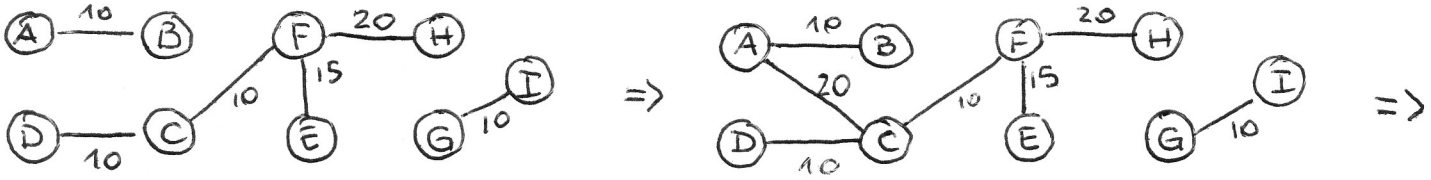
1	2	3	4	5	6	7	8	9	10
A	B	C	D	F	G	I	L	P	R

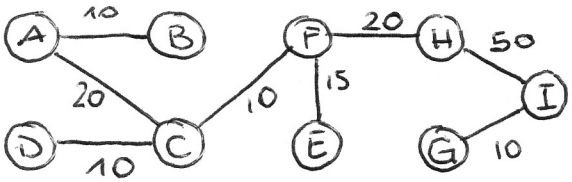
ESERCIZIO 2 KRUSKAL

- SI PARTE da un arco con peso minore, e man mano si aggiungono all'albero archi di peso minore di componenti connesse diverse









MST RISULTANTE

Esercizio 1)
Ordinare la seguente sequenza di chiavi tramite l'algoritmo *quicksort* indicando i passi eseguiti dall'algoritmo e il pivot utilizzato.

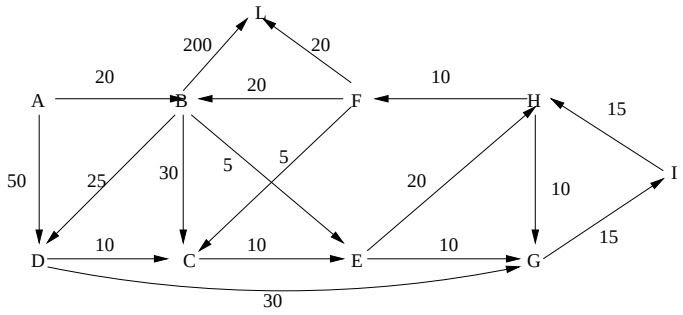
9	3	12	3	15	27	6	18	15	9

Esercizio 2)
Costruire le tabelle hash risultanti dall'inserimento dei seguenti valori: `58, 37, 67, 29, 78, 50` nell'assunzione di hashing con indirizzamento aperto e con funzione primaria di hash basata sul metodo della divisione, numero di celle nelle tabelle di hash m=10 e gestione delle collisioni tramite:

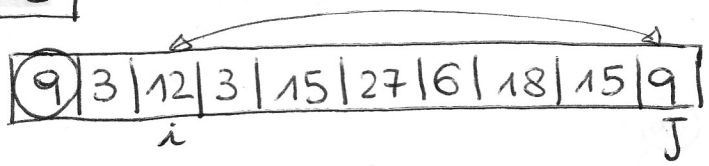
- ispezione lineare
- ispezione quadratica (con valori delle costanti $c_1 = 0, c_2 = 1$)
- hashing doppio (con $h_2(x) = 7 - (x \text{ MOD } 7)$)

Si richiede di compilare le tabelle allegate.

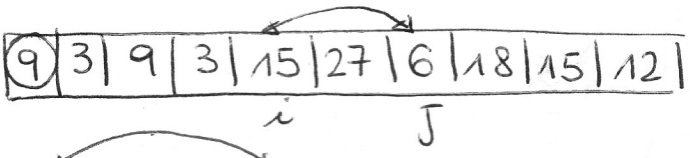
Esercizio 3)
Trovare e mostrare i *cammini minimi* (ed il relativo peso) dalla sorgente A ai vari nodi del grafo utilizzando l'algoritmo di Dijkstra, mostrando lo svolgimento passo passo dell'algoritmo.

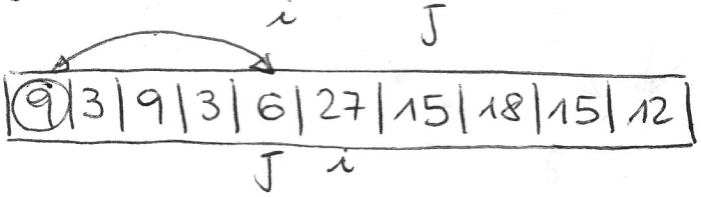


ERCIZIO 1 | QUICKSORT

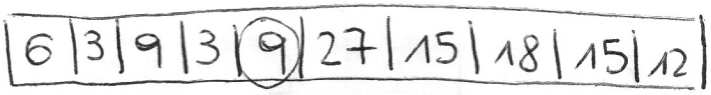


PIVOT: 9 (cerchiato)
i > PIVOT J ≤ PIVOT
(se così scambio i con J e procedo stesso modo)

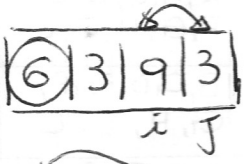




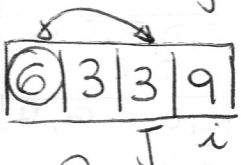
qnd la J precede la i si scambia con il pivot

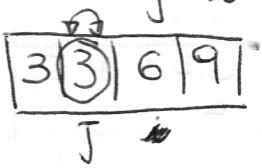


ora si esegue il quick a destra del pivot e poi a sinistra del pivot come se fossero 2 parti separate

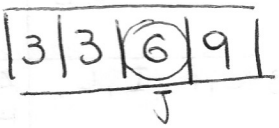


PIVOT: 6

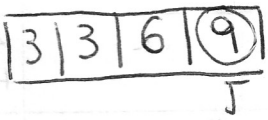




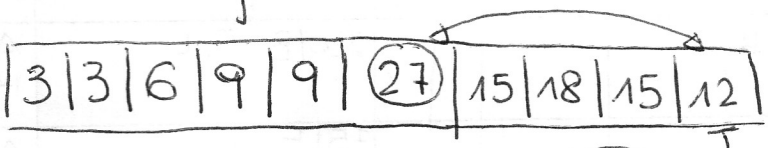
PIVOT: 3 (cerchiato) chiama ricorsivam. il quick sul 3



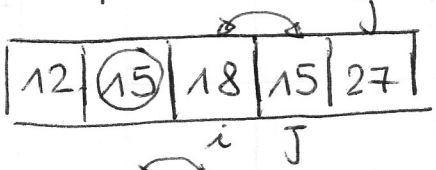
PIVOT: 6 chiama ricorsivam. il quick sul 6



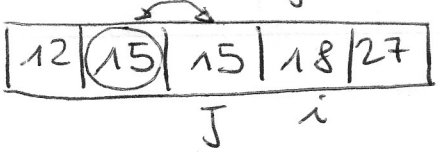
PIVOT: 9 chiama ricorsivam. il quick sul 9

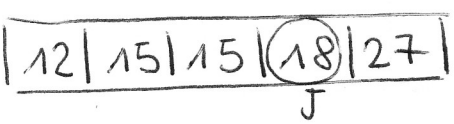


PIVOT: 27

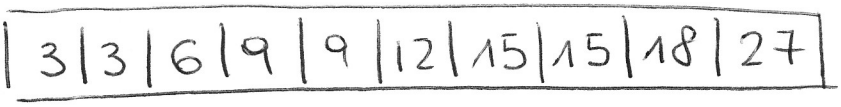


PIVOT: 15



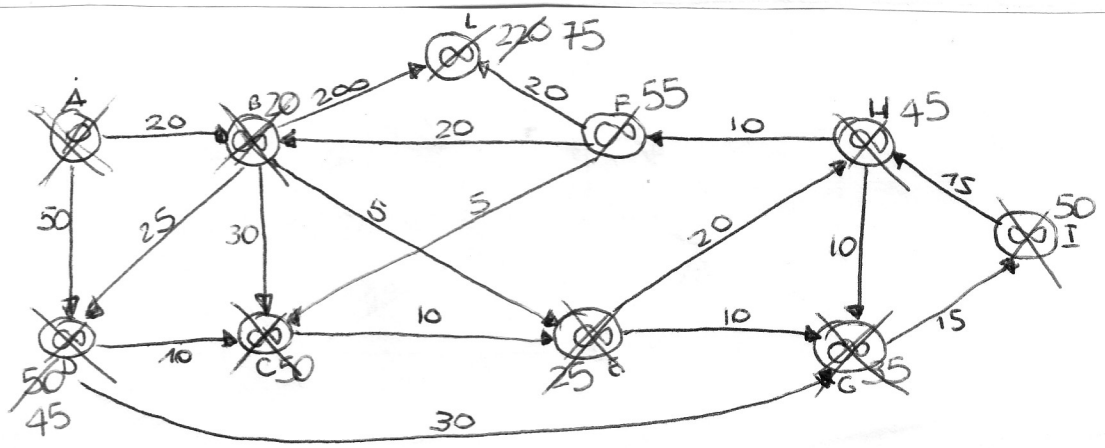


PIVOT: 18 poi 27



ARRAY ORDINATO

ESERCIZIO 3 DIJKSTRA



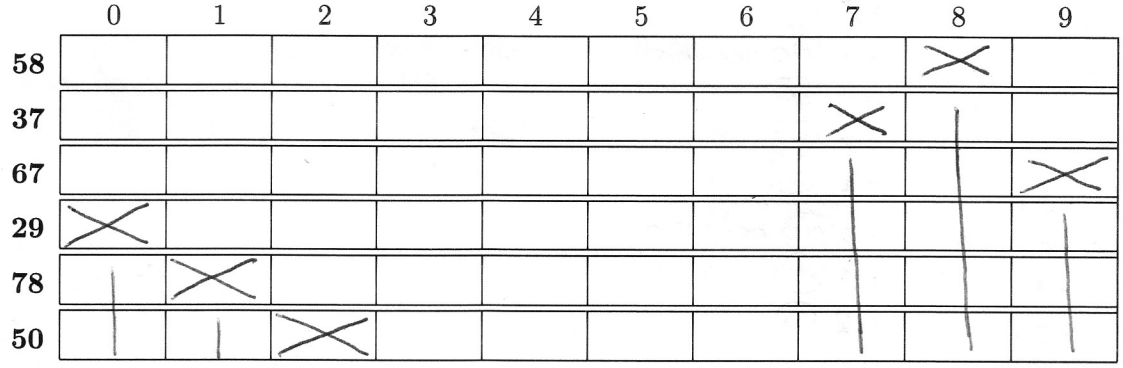
A	B	C	D	E	F	G	H	I	L	Q	S
ϕ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	A, B, C, D, E, F, G, H, I, L	ϕ
|	20 | A	∞ | N_{IL}	50 | A	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	∞ | N_{IL}	B, C, D, E, F, G, H, I, L	A
|	|	50 | B	45 | B	25 | B	∞ | N IL	∞ | N IL	∞ | N IL	∞ | N IL	220 | B	C, D, E, F, G, H, I, L	A, B
	|	50 | B	45 | B	|	∞ | N IL	35 | E	45 | E	∞ | N IL	220 | B	C, D, F, G, H, I, L	A, B, E
		50 | B	45 | B	|	∞ | N IL	|	45 | E	50 | G	220 | B	C, D, F, H, I, L	A, B, E, G
		50 | B	|		∞ | N IL		45 | E	50 | G	220 | B	C, F, H, I, L	A, B, E, G, D
		50 | B	|		55 | H		|	50 | G	220 | B	C, F, I, L	A, B, E, G, D, H
		|			55 | H		|	50 | G	220 | B	F, I, L	A, B, E, G, D, H, C
					55 | H			|	220 | B	F, L	A, B, E, G, D, H, C, I
					|				75 | F	L	A, B, E, G, D, H, C, I, F
										|	ϕ	A, B, E, G, D, H, C, I, F, L

$h'(k) = k \bmod m$

Esercizio 2)

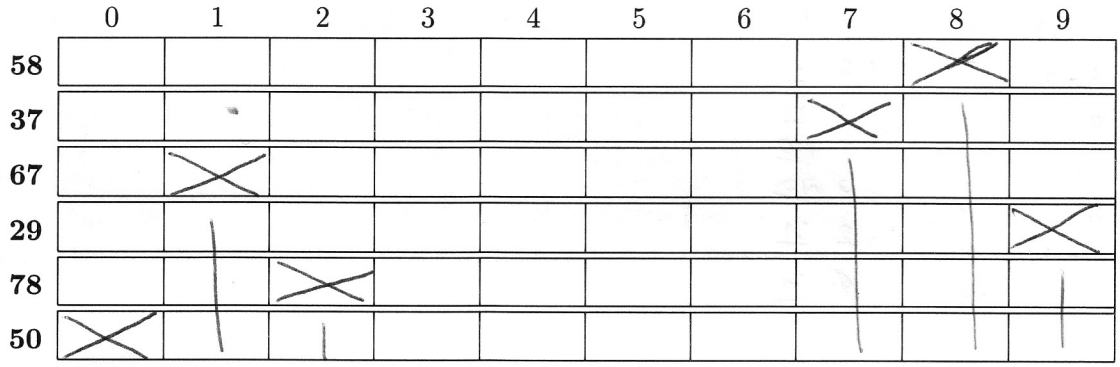
ISPEZIONE LINEARE $h(k,i) = (h'(k) + i) \bmod m$

Funzione hash ordinaria (h_1): Funzione (re)hash per ispezione:



ISPEZIONE QUADRATICA $h(k,i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$

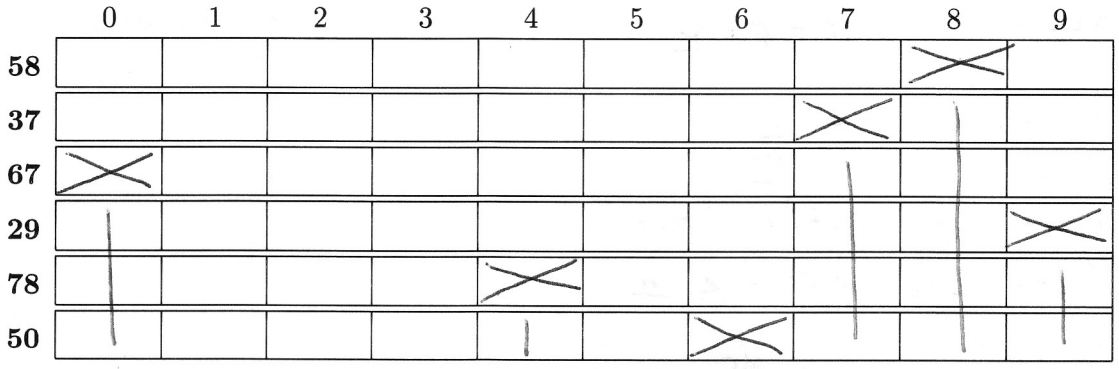
Funzione hash ordinaria (h_1): Funzione (re)hash per ispezione:



HASHING DOPPIO $h(k,i) = (h_1(k) + i\,h_2(k)) \bmod m$

Funzione hash ordinaria (h_1): Funzione hash ordinaria (h_2):

Funzione (re)hash per ispezione:



ISPEZIONE LINEARE

- $h(58,\emptyset) = (8+0) \bmod 10 = 8$ (SI)
- $h(37,\emptyset) = (7+0)$ " $= 7$ (SI)
- $h(67,\emptyset) = (7+0)$ " $= 7$ (NO) OCCUPATO
- $h(67,1) = (7+1)$ " $= 8$ (NO) "
- $h(67,2) = (7+2)$ " $= 9$ (SI)
- $h(29,\emptyset) = (9+0)$ " $= 9$ (NO) OCCUPATO
- $h(29,1) = (9+1)$ " $= \emptyset$ (SI)
- $h(78,\emptyset) = (8+0)$ " $= 8$ (NO) OCCUPATO
- $h(78,1) = (8+1)$ " $= 9$ (NO) "
- $h(78,2) = (8+2)$ " $= \emptyset$ (NO) "
- $h(78,3) = (8+3)$ " $= 1$ (SI)
- $h(50,\emptyset) = (0+0)$ " $= \emptyset$ (NO) OCCUPATO
- $h(50,1) = (0+1)$ " $= 1$ (NO) "
- $h(50,2) = (0+2)$ " $= 2$ (SI)

ISPEZIONE QUADRATICA $C_1 = 0$ $C_2 = 1$

- $h(58,\emptyset) = (8+0+0) \bmod 10 = 8$ SI
- $h(37,\emptyset) = (7+0)$ " $= 7$ SI
- $h(67,\emptyset) = (7+0)$ " $= 7$ NO
- $h(67,1) = (7+1)$ " $= 8$ NO
- $h(67,2) = (7+4)$ " $= 1$ SI
- $h(29,\emptyset) = (9+0)$ " $= 9$ SI
- $h(78,\emptyset) = (8+0)$ " $= 8$ NO
- $h(78,1) = (8+1)$ " $= 9$ NO
- $h(78,2) = (8+4)$ " $= 2$ SI
- $h(50,\emptyset) = (0+0)$ " $= \emptyset$ SI

HASHING DOPPIO $h_2(k) = 7 - (k \bmod 7)$

- $h(58,\emptyset) = (8+0) \bmod 10 = 8$ SI
- $h(37,\emptyset) = (7+0)$ " $= 7$ SI
- $h(67,\emptyset) = (7+0)$ " $= 7$ NO

$h(67,1) = [(7+7-1(67 \bmod 7)] \bmod 10 =$
$[(7+7-4)] \bmod 10 = \emptyset$ SI

- $h(29,\emptyset) = (9+0) \bmod 10 = 9$ SI
- $h(78,\emptyset) = (8+0)$ " $= 8$ NO

$h(78,1) = [(8+7-1(78 \bmod 7)] \bmod 10 =$
$[(8+7-1)] \bmod 10 = 4$ SI

- $h(50,\emptyset) = (0+0) \bmod 10 = \emptyset$ NO

$h(50,1) = [(0+7-1(50 \bmod 7)] \bmod 10 =$
$(7-1) \bmod 10 = 6$ SI