

- capitolo I
  - conversione numeri notazione posizionale in quantità [I.5, I.6, I.10, I.11, I.12]  
(notare che Fibonacci non dà un nome alle quantità oltre al migliaio)
    - input: 123
    - output: 1 centinaio  
2 decine  
3 unità
  - conversione numeri notazione posizionale in parole
    - input: 123
    - output: cento-venti-tre
  - conversione numeri romani <-> numeri arabi [I.7, I.8]
  - rappresentare i numeri con le mani [I.13]
  - tabelline dell'addizione e moltiplicazione [T.1]
- capitolo II
  - ~~○ algoritmo di moltiplicazione in colonna [II.2, II.10, II.13, II.16, II.19, II.28, II.35, II.41]~~
  - **ERRORE** [II.26]: alla quinta riga è "si moltiplichino 7 per 5"
  - **ERRORE** [II.18, II.23, II.26, II.27, II.33, II.34]: nei diagrammi il risultato ha degli zeri in più non presenti nei fattori
  - ~~○ prova del 9 [II.5 – II.7]~~
  - algoritmo di elevamento al quadrato [II.37 – II.40]  
(non so perché l'ho segnato, non c'è niente di speciale)
  - fare moltiplicazioni a mente [II.45 – II.49]
- capitolo III
  - ~~○ algoritmo di addizione in colonna [III.1]~~
  - ~~○ algoritmo di moltiplicazione "a scacchiera" [III.2 – III.4]~~
  - prova del 9 (dimostrazione geometrica) [III.9 – III.11]
  - sommare molti numeri [III.14]
- capitolo IV
  - ~~○ algoritmo di sottrazione in colonna [IV.1, IV.2]~~
- capitolo V
  - conversione frazioni ordinarie <-> frazioni multiple e altri tipi di frazione [V.3 – V.5]  
(“infilzare i rotti” da multiple a ordinarie e “traslatare i rotti” viceversa)
  - divisione in colonna per i numeri tabulati [V.6]
  - **ERRORE** [V.17]: nello schema i resti in nero vanno traslati di uno a destra
  - fare divisioni a mente [V.19]
  - divisione per 10 [V.21]
  - divisione per numeri primi [V.23 – V.25, V.34]

- prova con altri resti [V.38, V.66]
- **ERRORE** [V.41]:  $24059 = 6 \times 3 + 3 = 21 = 0 \pmod{7}$
- **ERRORE (?)**: lo schema a [V.40] dovrebbe essere posto a [V.44]
- ~~scomposizione in fattori primi [V.47]~~
- scomposizione di numeri pari [V.55]
- ~~divisione mediante scomposizione del dividendo [V.64, V.68]~~

- capitolo VI

- algoritmo di moltiplicazione di due numeri interi con una frazione semplice [VI.1.3 – VI.1.5, VI.1.9]
- una “forma normale” per le frazioni multiple [VI.1.12]
- semplificazione e algoritmo di Euclide [VI.1.4, VI.1.5]
- –
- algoritmo di moltiplicazione di due numeri interi con una frazione multipla [VI.2.1]
- algoritmo di moltiplicazione di due numeri interi con due frazioni semplici [VI.2.4]
- riduzione di frazioni multiple [VI.2.6]
- –
- algoritmo di addizione di frazioni (semplici) [VI.3.3 – VI.3.5]  
(notare che l’algoritmo che usa Fibonacci quando i denominatori hanno fattori in comune è sostanzialmente equivalente a quello usato oggi con il minimo comune multiplo)
- –
- algoritmo di moltiplicazione di frazioni [VI.6]  
(notare che la discussione delle parti precedenti può essere ricondotta a questa parte pensando gli interi come frazioni di denominatore 1)
- –
- riduzione di frazioni multiple [VI.7.2, VI.7.3]

- capitolo VII

- algoritmi di addizione, sottrazione e divisione di frazioni [VII.1.1, VII.1.2]
- –
- algoritmo del minimo comune multiplo [VII.2.14]
- –
- algoritmo di addizione di numeri interi con fratti [VII.4.1, VII.4.3, VII.4.10]
- –
- (rileggere parte [VII.5])
- –
- algoritmi di disgregazione [VII.6]

moltiplicazione

$$A = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10 + a_0$$

$$B = b_m \cdot 10^m + b_{m-1} \cdot 10^{m-1} + \dots + b_1 \cdot 10 + b_0$$

$$\begin{aligned} A \cdot B &= (a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0) \cdot \\ &\quad (b_m \cdot 10^m + b_{m-1} \cdot 10^{m-1} + \dots + b_1 \cdot 10^1 + b_0 \cdot 10^0) = \\ &= (a_n b_m) \cdot 10^{n+m} + (a_n b_{m-1} + a_{n-1} b_m) \cdot 10^{n+m-1} + \dots \\ &\quad + (a_1 b_0 + a_0 b_1) \cdot 10^1 + (a_0 b_0) \cdot 10^0 \end{aligned}$$

prova del 9

$$A = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10 + a_0$$

$$A \equiv a_n + a_{n-1} + \dots + a_1 + a_0 \pmod{9}$$

$$[A \cdot B]_9 = [ [A]_9 \cdot [B]_9 ]_9$$

$$[A + B]_9 = [ [A]_9 + [B]_9 ]_9$$