



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Appunti di Algoritmi e Strutture Dati

**a.a. 2017/2018**

Autore:  
**Timoty Granziero**

Repository:  
<https://github.com/Vashy/ASD-Notes>

28 luglio 2018

# Indice

<b>1</b>	<b>Lezione del 28/02/2018</b>	<b>5</b>
1.1	Problem Solving . . . . .	5
1.2	Cosa analizzeremo nel corso . . . . .	6
1.2.1	Approfondimento sul tempo di esecuzione $T(n)$ . . . . .	6
1.3	Problema dell'ordinamento (sorting) . . . . .	6
1.4	Insertion Sort . . . . .	7
1.4.1	Invarianti e correttezza . . . . .	8
<b>2</b>	<b>Lezione del 02/03/2018</b>	<b>9</b>
2.1	Modello dei costi . . . . .	9
2.2	Complessità di IS . . . . .	10
2.2.1	Caso migliore . . . . .	10
2.2.2	Caso peggiore . . . . .	10
2.2.3	Caso medio . . . . .	11
2.3	Divide et Impera . . . . .	11
2.4	Merge Sort . . . . .	11
2.4.1	Invarianti e correttezza . . . . .	13
<b>3</b>	<b>Lezione del 07/03/2018</b>	<b>15</b>
3.1	Approfondimento sull'induzione . . . . .	15
3.1.1	Induzione ordinaria . . . . .	15
3.1.2	Induzione completa . . . . .	15
3.2	Complessità di Merge Sort . . . . .	15
3.3	Confronto tra IS e MS . . . . .	17
<b>4</b>	<b>Lezione dell'08/03/2018</b>	<b>18</b>
4.1	Notazione asintotica . . . . .	18
4.1.1	Limite asintotico superiore . . . . .	19
4.1.2	Limite asintotico inferiore . . . . .	21
4.1.3	Limite asintotico stretto . . . . .	22
4.2	Metodo del limite . . . . .	23
4.3	Alcune proprietà generali . . . . .	23
<b>5</b>	<b>Lezione del 09/03/2018</b>	<b>24</b>
5.1	Complessità di un problema . . . . .	24
5.2	Esempio: limite inferiore per l'ordinamento basato su scambi . . . . .	24
5.3	Soluzione di ricorrenze . . . . .	25
5.3.1	Metodo di sostituzione . . . . .	26
<b>6</b>	<b>Lezione del 14/03/2018</b>	<b>28</b>

<b>7 Lezione del 15/03/2018</b>	<b>31</b>
7.1 Master Theorem . . . . .	31
7.1.1 Esercizi (Master Theorem) . . . . .	32
<b>8 Lezione del 21/03/2018</b>	<b>36</b>
8.1 Heapsort . . . . .	36
8.1.1 Max Heap . . . . .	37
8.2 Code con priorità . . . . .	40
<b>9 Lezione del 23/03/2018</b>	<b>43</b>
9.1 Quicksort . . . . .	43
9.1.1 Correttezza di Quicksort( $A, p, r$ ) . . . . .	44
9.1.2 Complessità di Quicksort . . . . .	45
<b>10 Lezione del 28/03</b>	<b>47</b>
10.1 Quicksort a tre partizioni . . . . .	47
10.2 Limite inferiore . . . . .	48
10.2.1 Albero di Decisioni . . . . .	48
10.3 Ordinamento in tempo lineare . . . . .	50
10.3.1 Counting Sort . . . . .	50
<b>11 Lezione del 29/03/2018</b>	<b>52</b>
11.1 Radix Sort . . . . .	52
11.1.1 Correttezza di RadixSort( $A, d$ ) . . . . .	52
11.2 Strutture dati elementari . . . . .	53
11.2.1 Tabelle Hash . . . . .	53
<b>12 Lezione del 04/04/2018</b>	<b>55</b>
12.1 Chaining . . . . .	55
12.1.1 Hashing uniforme semplice . . . . .	55
12.1.2 Funzioni Hash . . . . .	57
12.1.3 Hashing Universale . . . . .	58
<b>13 Lezione del 05/04/2018</b>	<b>59</b>
13.1 Open Addressing . . . . .	59
13.1.1 Hashing uniforme . . . . .	60
13.1.2 Funzioni di Hash . . . . .	60
<b>14 Lezione del 06/04/2018</b>	<b>64</b>
14.1 Max Heap con coda dinamica . . . . .	64

<b>15 Lezione del 26/04/2018</b>	<b>65</b>
15.1 Alberi Binari di Ricerca (ABR)	65
15.1.1 Visita simmetrica	66
15.1.2 Ricerca	66
15.1.3 Successore di un nodo	68
15.1.4 Inserimento	69
15.1.5 Eliminazione di un nodo	69
<b>16 Lezione del 27-28/04/2018</b>	<b>71</b>
16.1 Red-Black Trees	71
16.1.1 Complessità algoritmi RB-Trees	73
16.1.2 RB-Insert e RB-Delete	73
<b>17 Lezioni del 03-04/05/2018</b>	<b>78</b>
17.1 Arricchimento di Strutture Dati	78
17.1.1 Statistiche d'ordine	78
17.1.2 Teorema dell'aumento degli RB-Trees	81
17.1.3 Interval Trees	81
<b>18 Lezioni del 09-10-11/05/2018</b>	<b>83</b>
18.1 Programmazione Dinamica	83
18.1.1 Taglio delle aste	83
18.1.2 Prodotto di Matrici	88
18.1.3 Cammino minimo di un grafo*	92
<b>19 Lezioni del 16/05/2018</b>	<b>93</b>
19.1 Altri esempi di progr. dinamica	93
19.1.1 Longest Common Subsequence (LCS)	93
<b>Appendices</b>	<b>93</b>
<b>A Raccolta algoritmi</b>	<b>93</b>
A.1 Insertion Sort	93
A.2 Merge Sort	93
A.3 Insertion Sort ricorsivo	94
A.3.1 Correttezza di Insertion-Sort(A, j)	94
A.3.2 Correttezza di Insert(A, j)	95
A.4 CheckDup	95
A.4.1 Correttezza di DMerge(A,p,q,r)	96
A.5 SumKey	96
A.5.1 Correttezza di Sum(A, key)	97
A.6 Heapsort	99

---

A.7	Code con priorità . . . . .	100
<b>B</b>	<b>Esercizi</b>	<b>102</b>
B.1	Ricorrenze . . . . .	102
B.2	Esercizi svolti il 06/04/2018 . . . . .	102
B.3	Esercizio del 03/05/2018 . . . . .	105

## 19 Lezioni del 16/05/2018

### 19.1 Altri esempi di progr. dinamica

#### 19.1.1 Longest Common Subsequence (LCS)

Consideriamo le **basi azotate** del DNA: *Adenina*, *Citosina*, *Guanina*, *Timina* (A, C, G, T).

Date due stringhe

$$\begin{aligned}x_1 &: A C T A C C T G \\x_2 &: A T C A C C\end{aligned}$$

Definiamo

**eliot distance**:  $n$  passi per rendere uguali le due stringhe.

LCS tra  $x_1$  e  $x_2$ :  $A T A C C$

Consideriamo  $X = x_1 \dots x_m$ ,

e una sua sottosequenza  $x_{i_1} \dots x_{i_k}$  con  $i_1, \dots, i_k \in \{1, \dots, m\}$

**Problema** Date due sequenze

$$X = x_1 \dots x_m$$

$$Y = y_1 \dots y_n$$

Trovare  $W = w_1 \dots w_k$  tale che:

1.  $W$  è sottosequenza di  $X$  e  $Y$ ;
2. la lunghezza sia massima.

**Osservazioni**

1. C'è una sottosequenza comune;
2. LCS non è unica (e.g. consideriamo le stringhe  $AB$  e  $BA$ . La loro LCS è sia  $A$  che  $B$ )

$$LCS(X, Y) = \text{insieme di LCS di } X \text{ e } Y$$

3. Ricerca esaustiva impossibile.

**Sottostruttura ottima** Mi riduco a prefissi

$$X = x_1 \dots x_m$$

Prefissi  $X^k = x_1 \dots x_k$   $k \leq m$  (ad esempio  $X^0$  è  $\varepsilon$ )

Sia  $W = w_1 \dots w_k \in LCS(X, Y)$

Allora

1. Se  $x_m = y_n$  allora  $w_k = x_m = y_n$  e  $W^{k-1} \in LCS(X^{m-1}, Y^{n-1})$ ;
2. Se  $x_m \neq y_n$ 
  - (2a) se  $x_m \neq w_k$  allora  $W \in LCS(X^{m-1}, Y)$ ;
  - (2b) se  $y_n \neq w_k$  allora  $W \in LCS(X, Y^{n-1})$ .

### Dimostrazione

1.

$$X = x_1 \dots x_{m-1} x_m$$

$$Y = y_1 \dots y_{n-1} y_n$$

$$\circ w_k = x_m = y_n$$

Se  $w_k \neq x_m \Rightarrow W$  sottoseq. di  $X^{m-1}, Y^{n-1}$

Impossibile,  $Wx_m$  sarebbe sottoseq. di  $X$  e  $Y$  più lunga di  $W$

$$\circ W^{k-1} \in LCS(X^{m-1}, Y^{n-1})$$

$W^{k-1}$  se non ottima, esisterebbe  $W'$  sottoseq. di  $X^{m-1}$  e  $Y^{n-1}$  tale che  $|W'| > |W^{k-1}|$ , ottenendo  $W'x_m$  sottoseq. di  $X$  e  $Y$  con  $|W'x_m| > |W^{k-1}x_m| = |W|$ , **assurdo**.

2.  $x_m \neq y_n$

$$(2a) \ x_m \neq w_k$$

$\Rightarrow W$  è sottoseq. di  $X^{m-1}$

$W$  è sottoseq. di  $Y$

$W \in LCS(X^{m-1}, Y)$ ? (è ottima?)

Se no, esisterebbe  $W'$  sottoseq. di  $X^{m-1}$  e  $Y$

$$|W'| > |W|$$

Assurdo, dato che  $W'$  è anche sottoseq. di  $X$  e  $Y$ , inoltre  $W$  è ottima per  $X$  e  $Y$ ;

(2b) Duale.

**Espressione ricorsiva** Valore della soluzione ottima

**Dati**

$$X = x_1 \dots x_m$$

$$Y = y_1 \dots y_n$$

$$C[i, j] = \text{lunghezza di LCS di } X^i \text{ e } Y^j$$

$$C[i, j] = \begin{cases} 0 & \text{se } i = 0 \text{ o } j = 0 \\ C[i-1, j-1] + 1 & \text{se } i, j > 0 \text{ e } x_i = y_j \\ \max(C[i-1, j], C[i, j-1]) & \text{se } i, j > 0 \text{ e } x_i \neq y_j \end{cases}$$

$$b[i, j] = \begin{cases} \nwarrow & \text{se } x_i = y_j \text{ e } C[i, j] = 1 + C[i-1, j-1] \\ \leftarrow & \text{se } x_i \neq y_j \text{ e } C[i, j] = C[i-1, j] \\ \uparrow & \text{se } x_i \neq y_j \text{ e } C[i, j] = C[i, j-1] \end{cases}$$

**Pseudocodice**LCS( $X, Y$ )

```

1   $m = X.len, n = Y.len$ 
2  for  $i = 0$  to  $m$  //  $\Theta(m)$ 
3       $C[i, 0] = 0$ 
4  for  $j = 1$  to  $n$  //  $\Theta(n)$ 
5       $C[0, j] = 0$ 
6  for  $i = 1$  to  $m$  //  $\Theta(n \cdot m)$ 
7      for  $j = 1$  to  $n$ 
8          if  $X[i] = Y[j]$  // C'è un match
9               $C[i, j] = C[i-1, j-1]$ 
10              $b[i, j] = \nwarrow$ 
11         else
12             if  $C[i-1, j] \geq C[i, j-1]$ 
13                  $C[i, j] = C[i-1, j]$ 
14                  $b[i, j] = \leftarrow$ 
15             else
16                  $C[i, j] = C[i, j-1]$ 
17                  $b[i, j] = \uparrow$ 
18  return  $b, c$ 
```

PRINTLCS( $X, Y$ )

```

1   $b, C = \text{LCS}(X, Y)$ 
2  PRINTLCSREC( $X, b, m, n$ )
```



```
PRINTLCSREC( $X, b, i, j$ )
1  if  $i > 0$  and  $j > 0$ 
2      if  $b[i, j] = \nwarrow$ 
3          PRINTLCSREC( $X, b, i - 1, j - 1$ )
4          PRINT  $X[i]$ 
5      else if  $b[i, j] = \leftarrow$ 
6          PRINTLCSREC( $X, b, i - 1, j$ )
7      else
8          PRINTLCSREC( $X, b, i, j - 1$ )
```