

Algoritmi e Strutture Dati

moduli 2/3

Moreno Marzolla

Dipartimento di Informatica—Scienza e Ingegneria (DISI)

Università di Bologna

<https://www.moreno.marzolla.name/>

Copyright © 2010—2016, 2020, 2021
Moreno Marzolla, Università di Bologna, Italy
<https://www.moreno.marzolla.name/teaching/ASD/>



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Presentiamoci

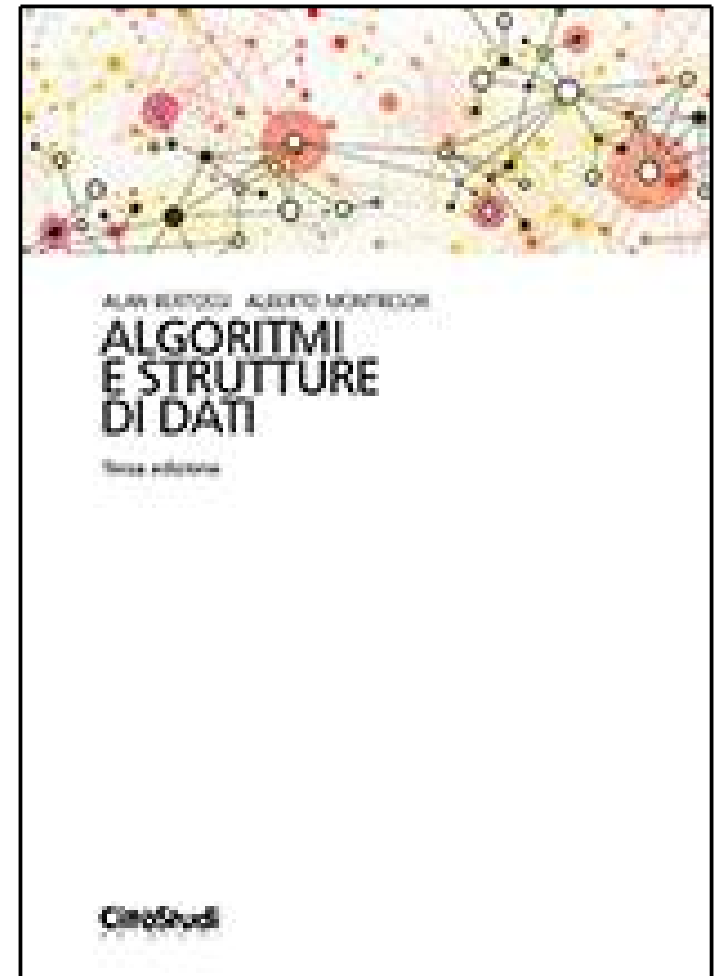
- Moduli 2/3 (II ciclo)
 - Moreno Marzolla, Saverio Giallorenzo
 - moreno.marzolla@unibo.it , saverio.giallorenzo2@unibo.it
 - <https://www.moreno.marzolla.name/>
 - Tutor: Tong Liu (modulo 1), Francesca Del Bonifro (moduli 2/3)
- Orario delle lezioni
 - Martedì 15:00–18:00
 - Mercoledì 15:00–18:00 (oppure 15:00–17:00 in caso di esercitazione)
- Ricevimento
 - Da concordare via mail

Sito web del modulo 3

- <https://www.moreno.marzolla.name/teaching/ASD>
 - Avvisi
 - Lucidi delle lezioni
 - Dispensa di esercizi svolti

Bibliografia

- Testo adottato
 - Alan Bertossi, Alberto Montresor, *Algoritmi e strutture di dati Terza Edizione*, 2014, Città Studi, ISBN: 9788825173956
- Testi di consultazione
 - Camil Demetrescu, Irene Finocchi, Giuseppe F. Italiano, *Algoritmi e strutture dati 2/ed*, 2008, McGraw-Hill, ISBN: 9788838664687
 - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduzione agli algoritmi e strutture dati 3/ed*, 2010, McGraw-Hill, ISBN: 9788838665158



Programma

- Analisi ammortizzata degli algoritmi
- Algoritmi di visita di grafi
- Tecniche algoritmiche
 - Divide-et-impera
 - Greedy
 - Programmazione dinamica
- Alberi di copertura (*spanning trees*)
- Cammini minimi
- Asserzioni e invarianti
- Macchine di Turing e teoria della calcolabilità
- Classi di complessità dei problemi (se avvanzerà tempo)

Prerequisiti

- Programmazione Internet + Lab. di prog. Internet
 - Algoritmi e Strutture Dati \neq Programmazione
 - In questo corso non si impara a programmare, perché dovrete già essere in grado di farlo
- Nozioni di base di algebra e analisi matematica
 - Sommatorie, polinomi, ordini di grandezza delle funzioni, disequazioni

Scopo del corso

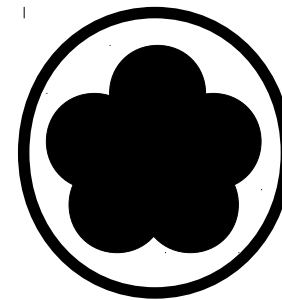
- **Contenuto**

- Una panoramica di problemi noti e loro soluzioni
- Elenco di algoritmi e strutture dati standard
- Come valutare l'efficienza degli algoritmi

- **Metodo**

- Principi e tecniche per risolvere problemi algoritmici
- Come risolvere nuovi problemi, applicando soluzioni note o “inventando” varianti alle soluzioni note

Scopo del corso



Modalità d'esame

- **Progetto** da svolgere **individualmente**
 - 4-5 algoritmi da progettare e realizzare in Java
 - 3 set di progetti diversi: uno per la sessione estiva (giugno-luglio), uno per quella autunnale (settembre), e uno per quella invernale (gennaio/febbraio 2022)
 - Specifiche disponibili circa un mese prima della consegna
 - Consegna tramite la piattaforma "Virtuale"
- Un progetto sufficiente consente l'accesso all'**orale** per la sessione cui si riferisce il progetto
- La prova orale include
 - Discussione del progetto
 - Domande su **tutti** gli argomenti svolti a lezione e durante le esercitazioni

Regole d'esame

- L'esame è un momento ufficiale e va affrontato con serietà
- Si sono verificati in passato casi di **gravi irregolarità**
 - Tali situazioni sono state (e saranno) sanzionate con la massima intransigenza
 - Esame annullato, da ripetere con nuovo progetto.
- L'esame orale è ugualmente un momento ufficiale
 - È sempre possibile rifiutare il voto e ritentare l'esame (nuovo progetto + nuovo orale)

Il vero significato della complessità degli algoritmi

Sottovettore di valore massimo

- Consideriamo un vettore $V[1..n]$ di $n \geq 1$ valori reali arbitrari
- Vogliamo individuare un sottovettore $V[i..j]$ non vuoto di V la somma dei cui elementi sia massima

3	-5	10	2	-3	1	4	-8	7	-6	-1
---	----	----	---	----	---	---	----	---	----	----

- Soluzione "di forza bruta": $O(n^3)$
- Esiste una soluzione $O(n)$
 - stay tuned...

Soluzione di “forza bruta” $O(n^3)$

```
real SommaMax1( real V[1..n] )
  real smax ← V[1];
  for integer i ← 1 to n do
    for integer j ← i to n do
      real s ← 0;
      for integer k ← i to j do
        s ← s + V[k];
      endfor
      if (s > smax) then
        smax ← s;
      endif
    endfor
  endfor
  return smax;
```

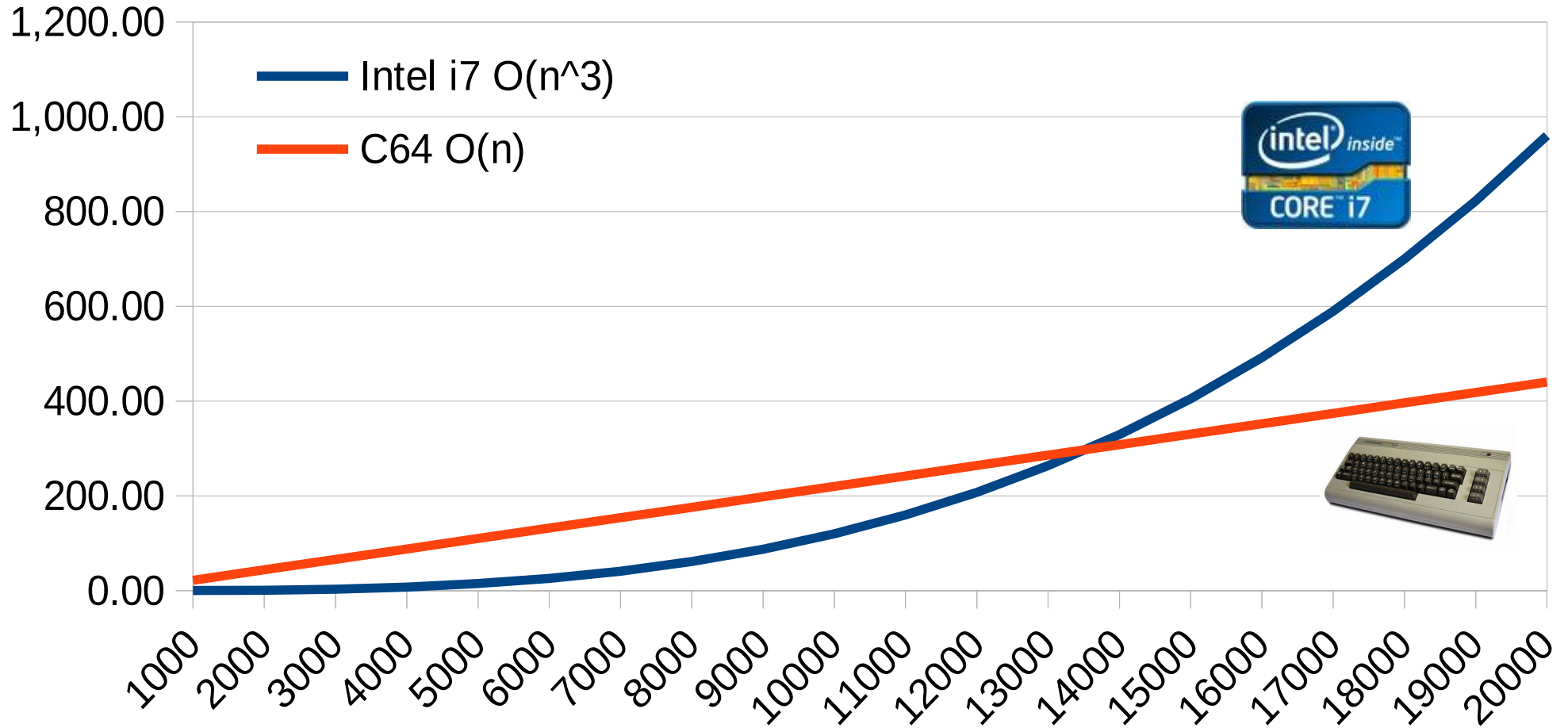
L'efficienza conta!

- Confrontiamo la soluzione $O(n^3)$ con una soluzione $O(n)$ (che vedremo più avanti) su due sistemi molto diversi
- **Algoritmo $O(n^3)$**
 - Ubuntu Linux 18.04
 - CPU: Intel i7 @ 3.6GHz
 - 16 GB RAM
 - Java (OpenJDK 11.0.10)
- **Algoritmo $O(n)$**
 - Commodore 64 (anno 1982)
 - CPU: MOS 6502 @ 1MHz
 - 64 KB RAM
 - Commodore BASIC



Sottovettore di somma massima

Tempi di esecuzione in secondi



Esercizi di ripasso

Vero o falso?

1. $1325 n^2 + 12n + 1 = O(n^3)$

2. $76 n^3 = O(n^3)$

3. $n^2 \log n = O(n^2)$

4. $3^n = O(2^n)$

5. $2^n = O(2^{n/2})$

6. $2^{n+100} = O(2^n)$

7. $\log n = O(n)$

8. $n = O(n \log n)$

9. $n^2 = O(n \log n)$

10. $\log(n^2) = O(\log n)$

11. $n(n+1) / 2 = O(n)$

Esercizio

- Determinare il costo asintotico dell'algorithmo seguente

```
ALGA( integer n ) → integer
  if ( n ≤ 1 ) then
    return 2*n;
  else
    integer a ← 2;
    for integer i ← 1 to n/2 do
      a ← a + 2 * i;
    endfor
    return ALGA( n/2 ) + ALGA( n/2 ) + a;
  endif
```

Esercizio

- Determinare il costo asintotico dell'algorithmo seguente

```
ALGB( integer n ) → integer
integer a ← 0;
Integer s, t;
for s ← 1 to n do
    for t ← s to n do
        a ← a + s + t;
    endfor
endfor
return a;
```

Strutture Dati

- Che differenza c'è tra `LinkedList` e `ArrayList` Java?

	<code>LinkedList</code>	<code>ArrayList</code>
Inserimento in testa		
Inserimento in coda		
Inserimento dopo un elemento di posizione/riferimento dati		
Cancellazione di un elemento di posizione/riferimento dati		
Accesso diretto al k -esimo elemento		

Caso ottimo / caso pessimo

- Consideriamo un albero binario di ricerca non bilanciato con n nodi
- Quale è il costo asintotico dell'operazione di ricerca
 - nel caso pessimo?
 - nel caso ottimo?