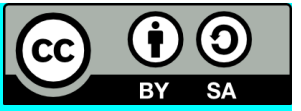
Algoritimi e Strutture Dati modulo 3

Moreno Marzolla
Dipartimento di Informatica—Scienza e Ingegneria (DISI)
Università di Bologna
https://www.moreno.marzolla.name/

Copyright © 2010—2016, 2020
Moreno Marzolla, Università di Bologna, Italy
http://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

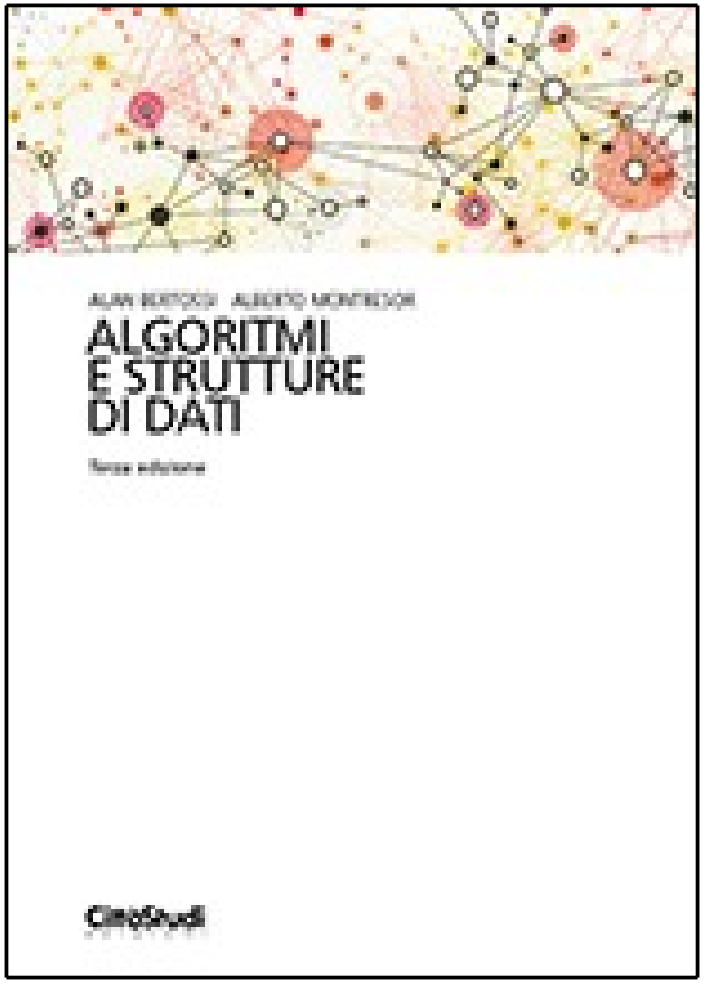
- Modulo 3 (II sem.)
  - Moreno Marzolla
  - moreno.marzolla@unibo.it
  - https://www.moreno.marzolla.name/
- Orario delle lezioni
  - Martedì 7/4/2020 ore 11:00–13:00
  - Giovedì 16/4/2020 ore 15:00–18:00
- A seguire:
  - Mercoledì ore 15:00—18:00
  - Giovedì ore 9:00—12:00
- Ricevimento
  - Da concordare via mail

Sito web del modulo 3

- https://www.moreno.marzolla.name/teaching/ASD
  - Avvisi
  - Lucidi delle lezioni
  - Dispensa di esercizi svolti
- Il materiale verrà inserito anche su IOL

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 del modulo 2

- Algoritmi su grafi
  - Alberi di copertura (*spanning trees*)
  - Cammini minimi
- Se avanzerà tempo
  - Asserzioni e invarianti
  - Macchine di Turing e teoria della calcolabilità
  - Classi di complessità dei problemi

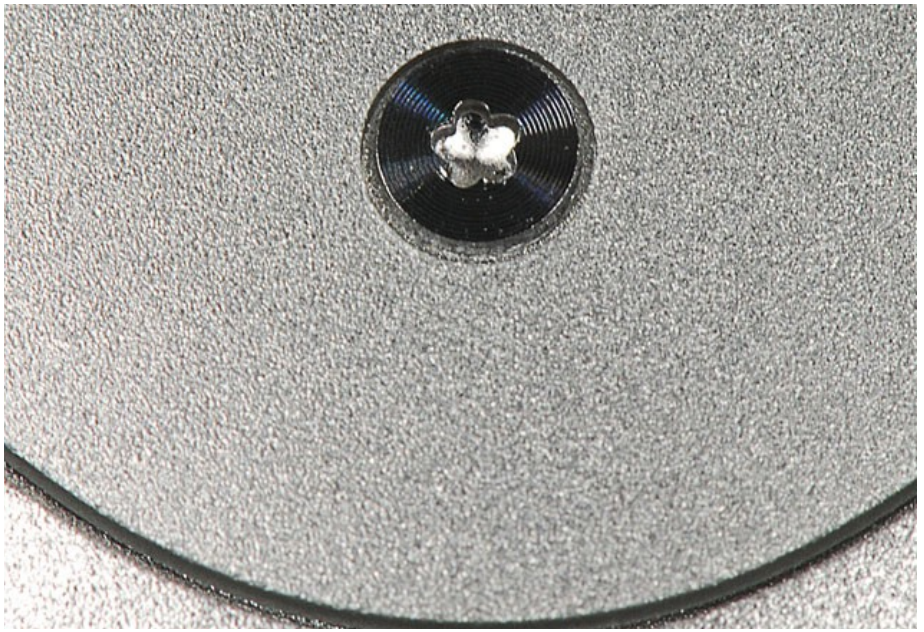
Prerequisiti

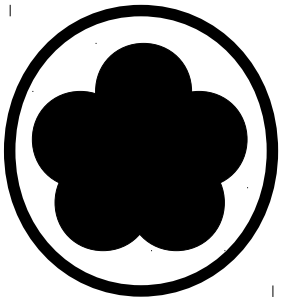
- Programmazione Internet + Lab. di prog. Internet
  - Algoritmi e Strutture Dati ≠ Programmazione
  - In questo corso non si impara a programmare, perché dovreste già essere in grado di farlo
- Nozioni di base di algebra e analisi matematica
  - Sommatorie, polinomi, ordini di grandezza delle funzioni,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

Scopo del corso

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



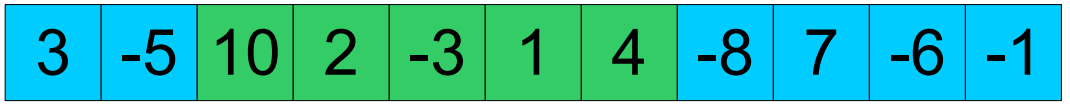


Modalità d'esame

- Sono quelle già descritte dai prof. Donatiello e Zavattaro
- Progetto di programmazione + discussioni

Una considerazione

- Ma qual è la vera differenza fra un algoritmo O(n) verso un algoritmo O(n^3)?
- Tocchiamola con mano riconsiderando il problema del sottovettore di somma massima
- Dato un un array $V[1..n]$ di $n > 0$ valori reali arbitrari vogliamo individuare un sottovettore non vuoto di V la somma dei cui elementi sia massima



Soluzione 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;
```

Soluzione O(*n*)

```
real SommaMax2( real V[1..n] )
   real S[1..n];
   S[1] ← V[1];
   integer imax ← 1; // indice val. max in S
   for integer i ← 2 to n do
      if ( S[i-1]+V[i] ≥ V[i] ) then
         S[i] ← S[i-1] + V[i];
      else
         S[i] ← V[i];
      endif
      if ( S[i] > S[imax] ) then
         imax ← i;
      endif
   endfor
   return S[imax];
```

L'efficienza conta!

- Confrontiamo i due algoritmi su due piattaforme hardware molto diverse
- Algoritmo O(n^3)
  - CPU: Intel i7 @ 3.6GHz
  - Ubuntu Linux 16.04
  - OpenJDK 11.0.6
- Algoritmo O(n)
  - Commodore 64 (anno 1982)
  - CPU: MOS 6502 @ 1MHz
  - Linguaggio: BASIC

Tempi di esecuzione