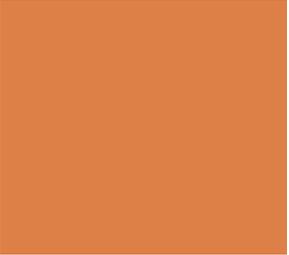


QUANTO SEI EFFICIENTE?

Irene Finocchi

Dipartimento di Informatica, Sapienza

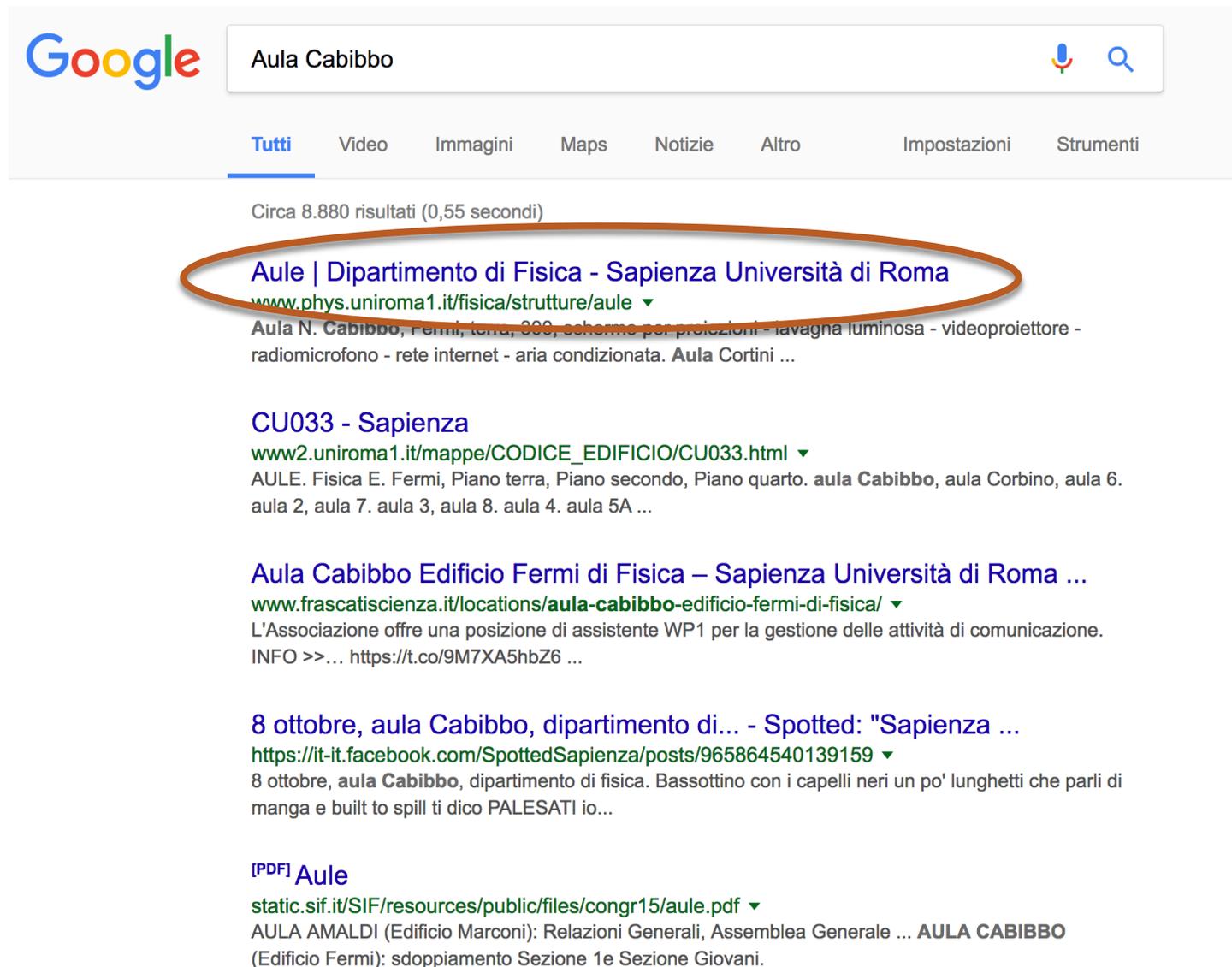
finocchi@di.uniroma1.it



Dov'è l'Aula Cabibbo?

La complessa semplicità della vita quotidiana

Dalla ricerca nel Web...



Google

Aula Cabibbo

Tutti Video Immagini Maps Notizie Altro Impostazioni Strumenti

Circa 8.880 risultati (0,55 secondi)

Aule | Dipartimento di Fisica - Sapienza Università di Roma
www.phys.uniroma1.it/fisica/strutture/aule ▼
Aula N. **Cabibbo**, Fermi, terra, 000, schermo per proiezioni - lavagna luminosa - videoproiettore - radiomicrofono - rete internet - aria condizionata. **Aula** Cortini ...

CU033 - Sapienza
www2.uniroma1.it/mappe/CODICE_EDIFICIO/CU033.html ▼
AULE. Fisica E. Fermi, Piano terra, Piano secondo, Piano quarto. **aula Cabibbo**, aula Corbino, aula 6. aula 2, aula 7. aula 3, aula 8. aula 4. aula 5A ...

Aula Cabibbo Edificio Fermi di Fisica – Sapienza Università di Roma ...
www.frascaticienza.it/locations/aula-cabibbo-edificio-fermi-di-fisica/ ▼
L'Associazione offre una posizione di assistente WP1 per la gestione delle attività di comunicazione. INFO >>... <https://t.co/9M7XA5hbZ6> ...

8 ottobre, aula Cabibbo, dipartimento di... - Spotted: "Sapienza ...
<https://it-it.facebook.com/SpottedSapienza/posts/965864540139159> ▼
8 ottobre, **aula Cabibbo**, dipartimento di fisica. Bassottino con i capelli neri un po' lunghetti che parli di manga e built to spill ti dico PALESATI io...

[PDF] Aule
static.sif.it/SIF/resources/public/files/congr15/aule.pdf ▼
AULA AMALDI (Edificio Marconi): Relazioni Generali, Assemblea Generale ... **AULA CABIBBO** (Edificio Fermi): sdoppiamento Sezione 1e Sezione Giovani.

... ai navigatori satellitari

The screenshot shows a navigation app interface. On the left, there is a menu with icons for car, bus, walking, bicycle, and airplane. Below the menu, the start and end points are listed: Via Salaria, 113, 00198 Roma and Piazzale della Minerva, 00185 Roma. The departure time is set to 'Partenza adesso'. Below this, there are three route options:

- tramite Viale dell'Università: 9 min, 2,9 km. Percorso migliore.
- tramite Viale Castro Pretorio: 9 min, 2,9 km. Questo percorso ha strade a traffico limitato o private.
- tramite Corso d'Italia: 9 min, 4,3 km.

The map on the right shows the route highlighted in blue and red. A small box on the map indicates the estimated time and distance for the selected route: 9 min, 2,9 km. The map also shows various landmarks and street names in Rome, including Via Salaria, Viale dell'Università, and Piazzale della Minerva.

Due problemi, tre algoritmi

□ Ricercare nel Web

- ▣ Se interrogo un motore di ricerca (Google), quali sono le risposte più rilevanti per la mia interrogazione?
 - Esempio: pagine importanti relative all'argomento "Università Sapienza"
- ▣ L'algoritmo PageRank calcola l'importanza delle pagine Web
- ▣ Un altro algoritmo **ordina le risposte** per importanza decrescente

□ Ricercare un percorso

- ▣ Qual è il miglior tragitto, a piedi o in autobus, tra l'Aula Cabibbo e il Dipartimento di informatica?
- ▣ Un algoritmo calcola un cammino con tempo di percorrenza minimo esaminando strade e percorsi dei mezzi pubblici



Ma cos'è un algoritmo?

Cosa dice il vocabolario?



The image shows a screenshot of the Treccani.it website. At the top, the logo "Treccani.it" is displayed in a large, serif font, with "L'ENCICLOPEDIA ITALIANA" written below it in a smaller, sans-serif font. Below the logo, there are four navigation links: "Enciclopedia", "Vocabolario", "Sinonimi", and "Dizionario Biografico degli Italiani". A search bar is positioned in the center, containing the text "algoritmo" and a red "Cerca" button. The background of the website features two images: a close-up portrait of a bearded man wearing a crown on the left, and a historical scene of a group of people on horseback in a mountainous landscape on the right. At the bottom of the page, there are five categories: "Scienze sociali e Storia", "Arte, Lingua e Letteratura", "Sport e Tempo libero", "Scienze naturali e Matematiche", and "Tecnologia e Scienze applicate".

Treccani.it
L'ENCICLOPEDIA ITALIANA

[Enciclopedia](#) | [Vocabolario](#) | [Sinonimi](#) | [Dizionario Biografico degli Italiani](#)

algoritmo [Cerca](#)

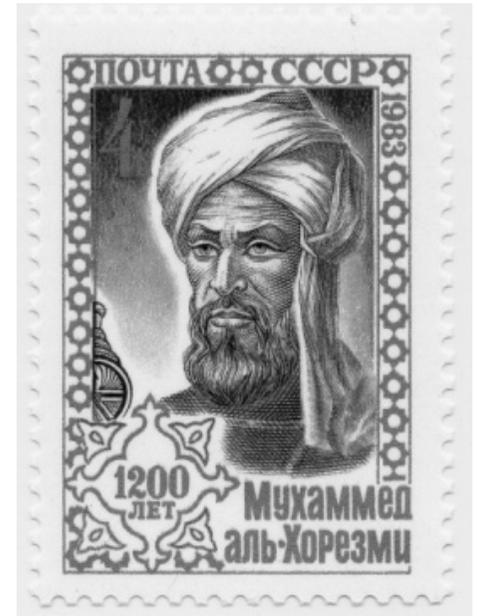
Scienze sociali e Storia | Arte, Lingua e Letteratura | Sport e Tempo libero | Scienze naturali e Matematiche | Tecnologia e Scienze applicate

algoritmo (ant. **algorismo**) s. m. [dal lat. mediev. *algorithmus* o *algorismus*, dal nome d'origine, *al-Khuwārizmī*, del matematico arabo Muḥammad ibn Mūsa del 9° sec. (così chiamato perché nativo di Khwarizm, regione dell'Asia Centrale)]. –

- 1.** Termine che indicò nel medioevo i procedimenti di calcolo numerico fondati sopra l'uso delle cifre arabe. Nell'uso odierno, anche con riferimento all'uso dei calcolatori, qualunque schema o procedimento matematico di calcolo; più precisamente, un procedimento di calcolo esplicito e descrivibile con un numero finito di regole che conduce al risultato dopo un numero finito di operazioni, cioè di applicazioni delle regole. In partic., *a. euclideo*, metodo per determinare il massimo comune divisore di due numeri interi a e b , basato su divisioni successive.
- 2.** In informatica, insieme di istruzioni che deve essere applicato per eseguire un'elaborazione o risolvere un problema.
- 3.** In logica matematica, qualsiasi procedimento «effettivo» di computo di una funzione o di decisione di un insieme (o predicato), cioè qualsiasi procedimento che consenta, con un numero finito di passi eseguiti secondo un insieme finito di regole esplicite, di ottenere il valore della funzione per un dato argomento, o di decidere se un dato individuo appartiene all'insieme (o soddisfa il predicato).

Etimologia

- Abdallah Mohamed Ibn Musa **al-Khuwarizmi** al-Magusi
- **Importante matematico, astronomo e geografo del mondo arabo-islamico (780-850 ca)**
- Contribuì alla diffusione del sistema di numerazione posizionale decimale nei paesi islamici (e da lì in Europa)
- Fu tra i primi a far riferimento al concetto di **procedimento per risolvere un problema in un numero finito di passi**
- Il suo nome, trascritto in latino e forse anche influenzato dal greco **αριθμος** (numero), è passato ad identificare le regole di calcolo



algoritmo (ant. **algorismo**) s. m. [dal lat. mediev. *algorithmus* o *algorismus*, dal nome d'origine, *al-Khuwārizmī*, del matematico arabo Muḥammad ibn Mūsa del 9° sec. (così chiamato perché nativo di Khwarizm, regione dell'Asia Centrale)]. –

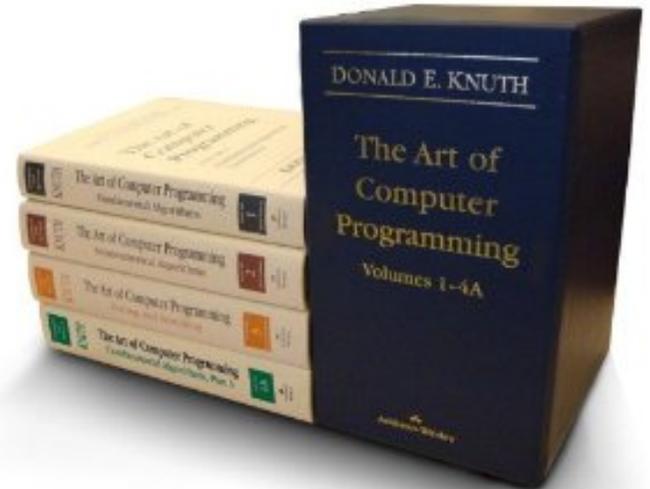
1. Termine che indicò nel medioevo i procedimenti di calcolo numerico fondati sopra l'uso delle cifre arabe. Nell'uso odierno, anche con riferimento all'uso dei calcolatori, qualunque schema o procedimento matematico di calcolo; più precisamente, un procedimento di calcolo esplicito e descrivibile con un numero finito di regole che conduce al risultato dopo un numero finito di operazioni, cioè di applicazioni delle regole. In partic., *a. euclideo*, metodo per determinare il massimo comune divisore di due numeri interi a e b , basato su divisioni successive.

2. In informatica, insieme di istruzioni che deve essere applicato per eseguire un'elaborazione o risolvere un problema.

3. In logica matematica, qualsiasi procedimento «effettivo» di computo di una funzione o di decisione di un insieme (o predicato), cioè qualsiasi procedimento che consenta, con un numero finito di passi eseguiti secondo un insieme finito di regole esplicite, di ottenere il valore della funzione per un dato argomento, o di decidere se un dato individuo appartiene all'insieme (o soddisfa il predicato).

The Art of Computer Programming

- La Bibbia degli algoritmi
- Opera scritta da **Donald E. Knuth** (opera magna, ancora incompleta!)
- **American Scientist**: l'ha inclusa tra i "100 or so Books that shaped a Century of Science" (XX secolo)
- **Bill Gates**: *"If you think you're a really good programmer, read (Knuth's) Art of Computer Programming... You should definitely send me a résumé if you can read the whole thing."*



Algoritmi secondo Knuth

- Un algoritmo, dato un **input**, genera un **output** proseguendo per passi successivi (sequenza di operazioni elementari)
- Tre proprietà:
 - **Definitezza**: ogni passo deve essere definito in modo chiaro e non ambiguo, dando luogo alla stessa sequenza di operazioni, indipendentemente da chi lo esegue e in quale momento
 - **Finitezza**: l'algoritmo deve terminare dopo l'esecuzione di un numero finito di passi
 - **Effettività**: deve essere effettivamente eseguibile, ovvero ogni operazione deve essere sufficientemente basilare da poter essere eseguita "a carta e penna" in una quantità finita di tempo



Esempio 1: somma

Problema

Dato n , calcolare la somma S dei primi n numeri interi:

$$S = \sum_{i=1}^n i$$

Input: n

Output: s

Due soluzioni

Algoritmo 1

□ Passo 1:

$$S = 0$$

□ Passo 2:

per ogni i da 1 a n :

$$S = S + i \text{ (aggiunge } i \text{ a } S)$$

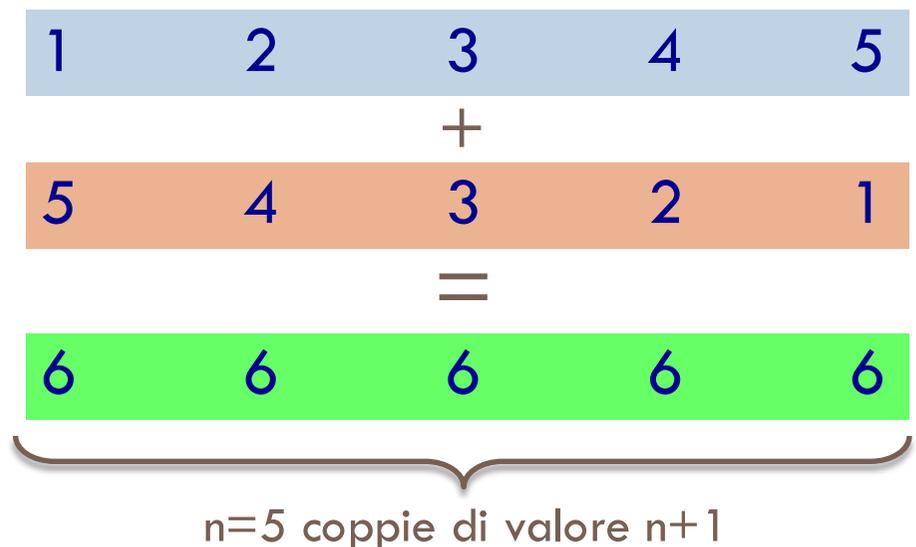
□ Passo 3:

restituisce S in output

Algoritmo 2

□ Passo 1:

restituisce in output il
valore $n(n+1)/2$



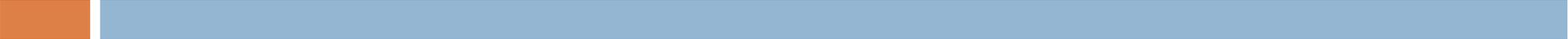
Una lente d'ingrandimento



*Se è vero che un problema non si capisce a fondo finché non lo si deve insegnare a qualcun altro, a maggior ragione **nulla è compreso in modo più approfondito di ciò che si deve insegnare ad una macchina, ovvero di ciò che va espresso tramite un algoritmo.***

Donald Knuth

Esempio 2: ricerca



Problema

Trovare il numero di telefono di Irene Finocchi
nell'elenco telefonico di Roma

Input: elenco telefonico, "Irene Finocchi"

Output: numero di telefono, se presente in elenco

Due soluzioni

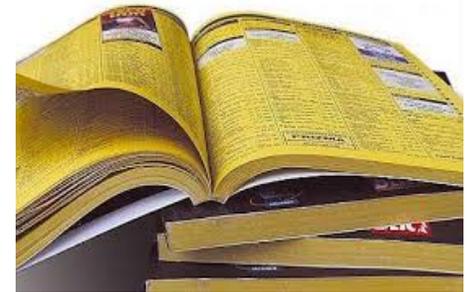
Algoritmo 1

- scorri l'elenco dalla prima pagina, leggendo tutti i nomi degli abbonati finché non trovi "Finocchi Irene"

Quale approccio è più rapido?

Algoritmo 2

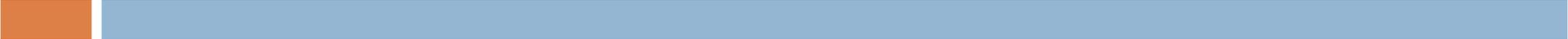
- apri l'elenco più o meno a metà e confronta "Finocchi Irene" con il nome all'inizio della pagina
- prosegui nell'elenco nella metà di destra o di sinistra a seconda dell'esito del confronto





Il tempo è tiranno

Algoritmi più o meno efficienti



- Operazioni elementari: operazioni di confronto, di assegnazione, aritmetico/logiche...
- Contiamo il numero di operazioni elementari compiute da un algoritmo
- Se lo ripetiamo per diverse dimensioni dell'input, otteniamo una **funzione** che misura l'efficienza dell'algoritmo
- Quanto meno la funzione cresce al crescere della dimensione dell'input, tanto più l'algoritmo verrà considerato efficiente

Esempio 1: somma

Algoritmo 1

□ Passo 1:

$$S = 0$$

□ Passo 2:

per ogni i da 1 a n :

$$S = S + i \text{ (aggiunge } i \text{ a } S)$$

□ Passo 3:

restituisce S in output

≈ n operazioni elementari

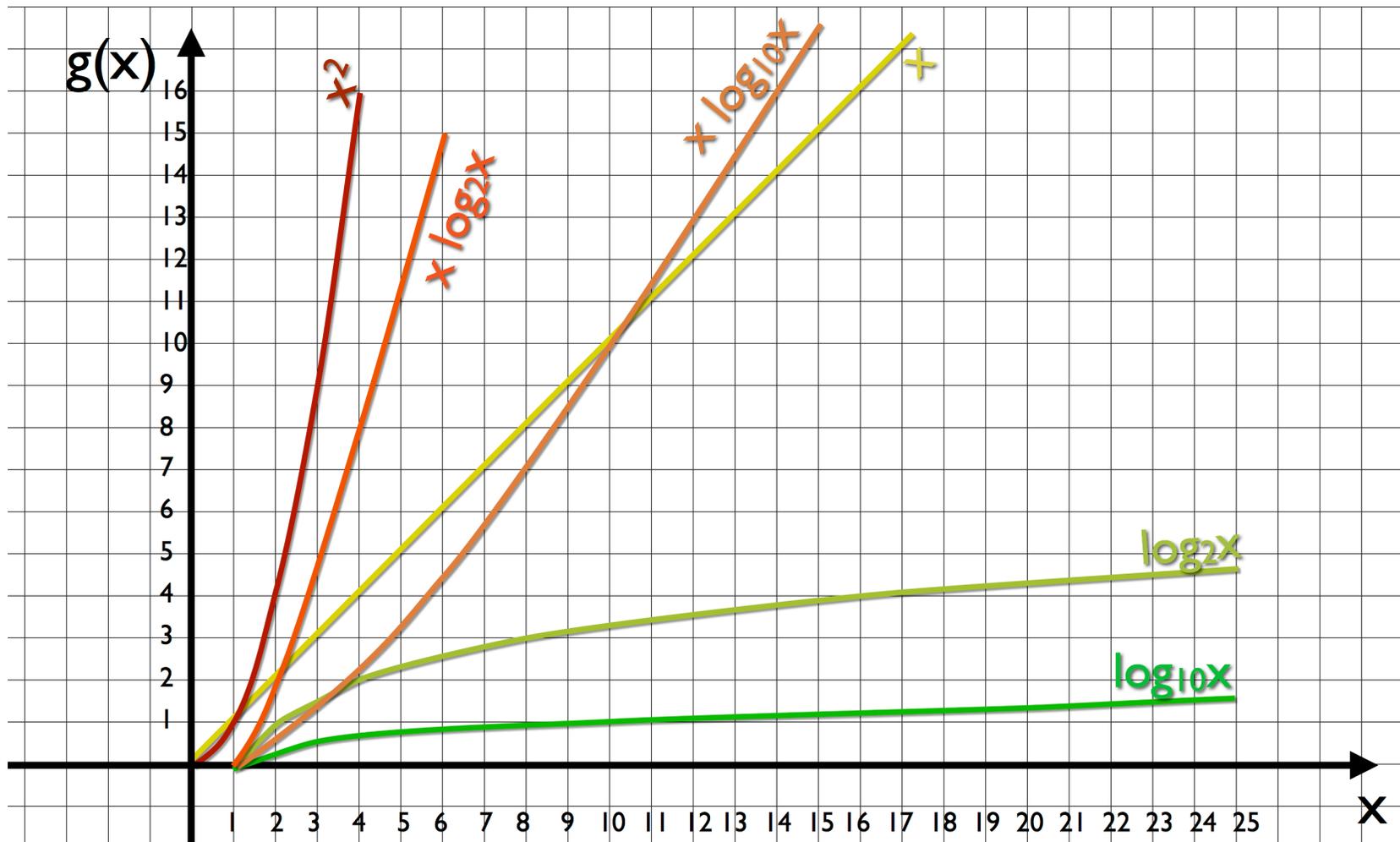
Algoritmo 2

□ Passo 1:

restituisce in output il
valore $n(n+1)/2$

3 operazioni elementari

Funzioni tipiche



Quante operazioni al crescere di n?

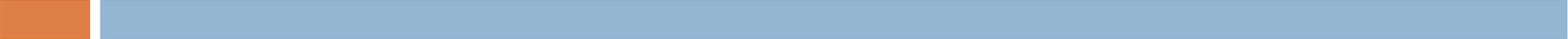
n	$\log_{10} n$	n	$n \log_{10} n$	n^2	2^n
10	1	10	10	100	$\approx 10^3$
100	2	100	200	10 000	$\approx 1.26 \times 10^{30}$
1000	3	1000	3000	1 000 000	$\approx 1.07 \times 10^{301}$
10000	4	10000	40 000	100 000 000	?

Quanto tempo al crescere di n?

Se una operazione elementare richiedesse **1 secondo...**

n	$\log_{10} n$	n	$n \log_{10} n$	n^2	2^n
10	1 s	10 s	10 s	100 s	16 m
100	2 s	100 s	3 m	3 h	4×10^{20} secoli
1000	3 s	16 m	50 m	11 giorni	?
10000	4 s	3 h	12 h	3 anni	?

Ancora sulla ricerca: algoritmo 1



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

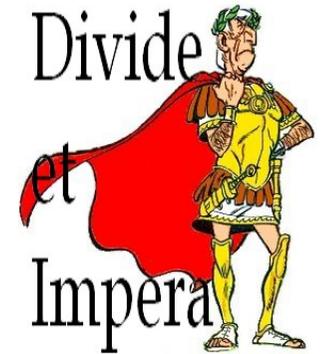
scorri i numeri da sinistra verso destra, finché non trovi quello cercato o un numero più grande

- Al più n operazioni elementari (confronti)
- Potrebbero essere meno, ma gli informatici ragionano in modo **pessimista**

Ancora sulla ricerca: algoritmo 2

Ricerca binaria (o dicotomica)

Cerchiamo 33



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

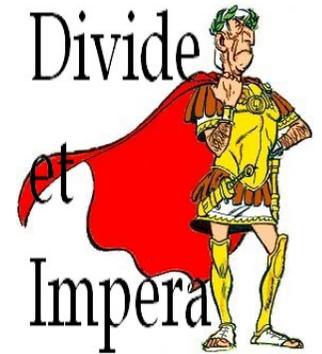
1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ancora sulla ricerca: algoritmo 2

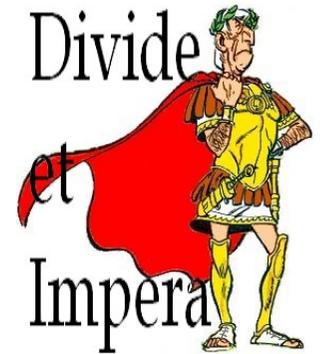
16=n elementi tra cui cercare



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ancora sulla ricerca: algoritmo 2

$8 = n/2$ elementi tra cui cercare

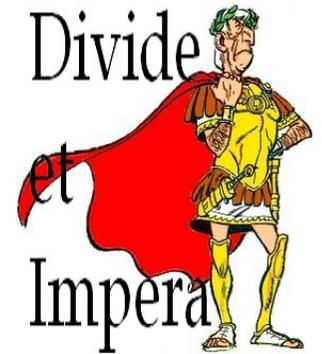


1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ancora sulla ricerca: algoritmo 2

$4 = n/2^2$ elementi tra cui cercare



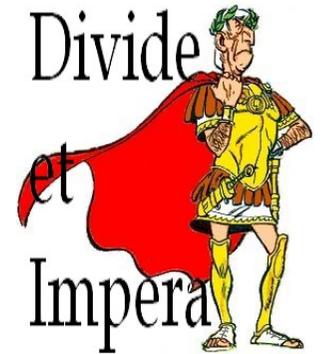
1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ancora sulla ricerca: algoritmo 2

$2 = n/2^3$ elementi tra cui cercare



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

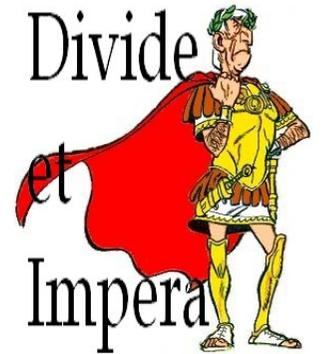
1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ancora sulla ricerca: algoritmo 2

$1 = n/2^4$ elementi tra cui cercare



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

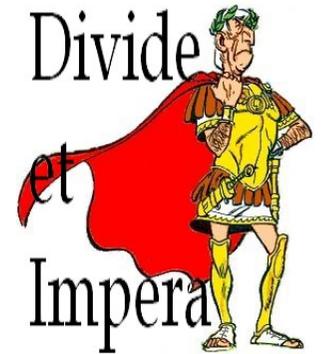
1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Analisi: $\approx \log_2 n$ passi

Per quale k si ha $n/2^k=1$?

$$2^k=n \iff k=\log_2 n$$



1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

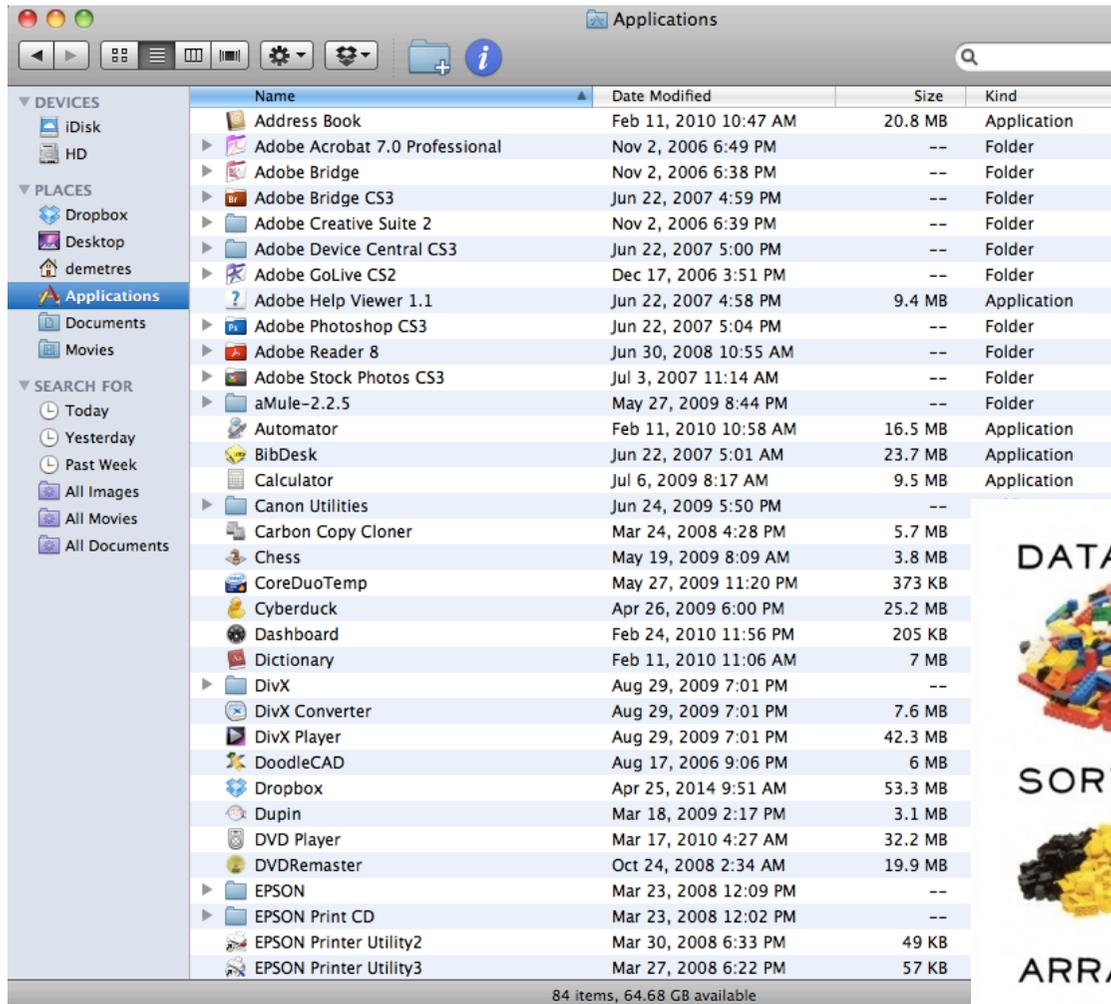
1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

1	3	4	5	7	8	11	15	28	31	33	50	62	68	70	91
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----

Ordinare dati: un problema importante



Name	Date Modified	Size	Kind
Address Book	Feb 11, 2010 10:47 AM	20.8 MB	Application
Adobe Acrobat 7.0 Professional	Nov 2, 2006 6:49 PM	--	Folder
Adobe Bridge	Nov 2, 2006 6:38 PM	--	Folder
Adobe Bridge CS3	Jun 22, 2007 4:59 PM	--	Folder
Adobe Creative Suite 2	Nov 2, 2006 6:39 PM	--	Folder
Adobe Device Central CS3	Jun 22, 2007 5:00 PM	--	Folder
Adobe GoLive CS2	Dec 17, 2006 3:51 PM	--	Folder
Adobe Help Viewer 1.1	Jun 22, 2007 4:58 PM	9.4 MB	Application
Adobe Photoshop CS3	Jun 22, 2007 5:04 PM	--	Folder
Adobe Reader 8	Jun 30, 2008 10:55 AM	--	Folder
Adobe Stock Photos CS3	Jul 3, 2007 11:14 AM	--	Folder
aMule-2.2.5	May 27, 2009 8:44 PM	--	Folder
Automator	Feb 11, 2010 10:58 AM	16.5 MB	Application
BibDesk	Jun 22, 2007 5:01 AM	23.7 MB	Application
Calculator	Jul 6, 2009 8:17 AM	9.5 MB	Application
Canon Utilities	Jun 24, 2009 5:50 PM	--	Folder
Carbon Copy Cloner	Mar 24, 2008 4:28 PM	5.7 MB	Application
Chess	May 19, 2009 8:09 AM	3.8 MB	Application
CoreDuoTemp	May 27, 2009 11:20 PM	373 KB	Application
Cyberduck	Apr 26, 2009 6:00 PM	25.2 MB	Application
Dashboard	Feb 24, 2010 11:56 PM	205 KB	Application
Dictionary	Feb 11, 2010 11:06 AM	7 MB	Application
DivX	Aug 29, 2009 7:01 PM	--	Folder
DivX Converter	Aug 29, 2009 7:01 PM	7.6 MB	Application
DivX Player	Aug 29, 2009 7:01 PM	42.3 MB	Application
DoodleCAD	Aug 17, 2006 9:06 PM	6 MB	Application
Dropbox	Apr 25, 2014 9:51 AM	53.3 MB	Application
Dupin	Mar 18, 2009 2:17 PM	3.1 MB	Application
DVD Player	Mar 17, 2010 4:27 AM	32.2 MB	Application
DVDRemaster	Oct 24, 2008 2:34 AM	19.9 MB	Application
EPSON	Mar 23, 2008 12:09 PM	--	Folder
EPSON Print CD	Mar 23, 2008 12:02 PM	--	Folder
EPSON Printer Utility2	Mar 30, 2008 6:33 PM	49 KB	Application
EPSON Printer Utility3	Mar 27, 2008 6:22 PM	57 KB	Application

84 items, 64.68 GB available



DATA



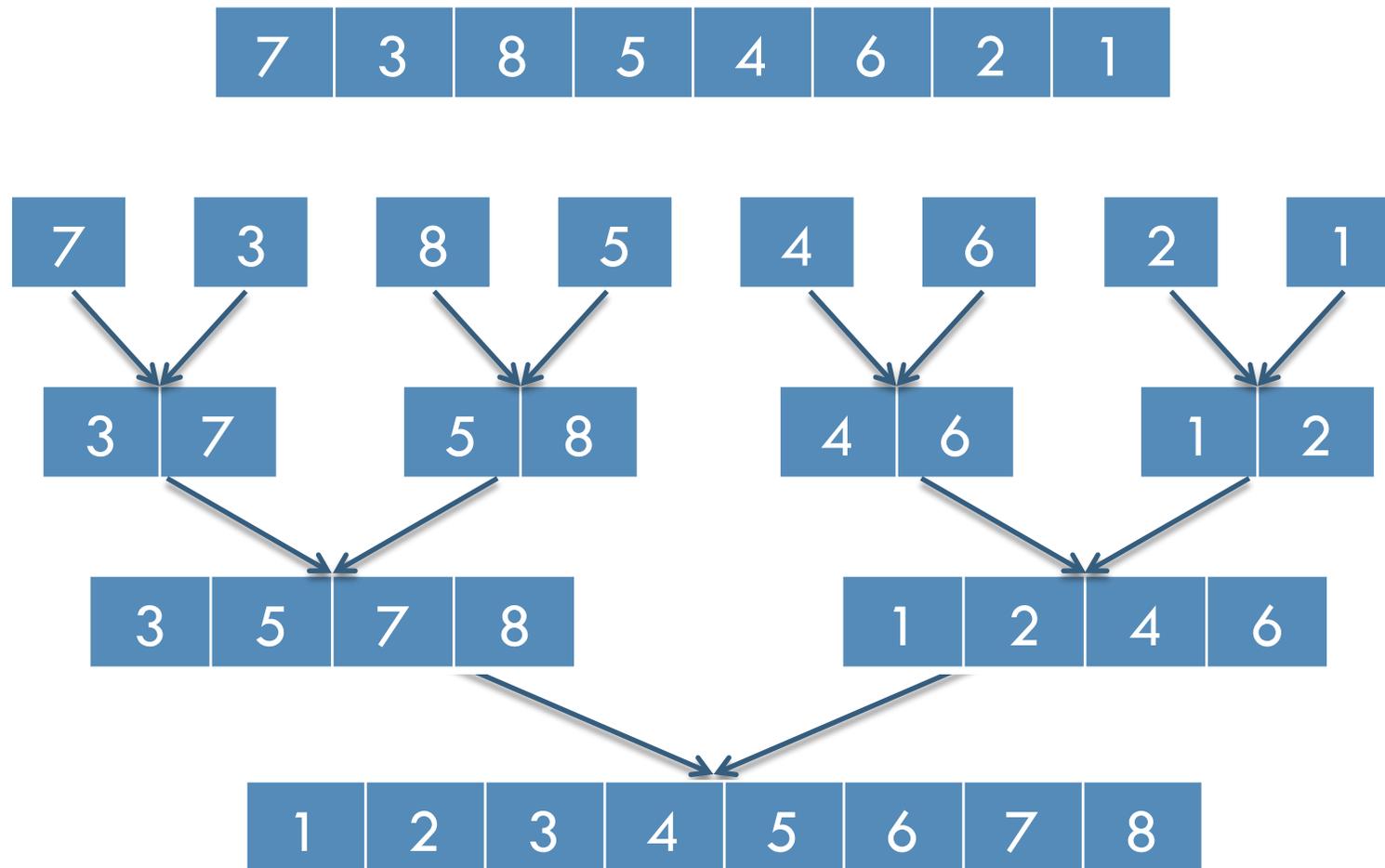
SORTED



ARRANGED



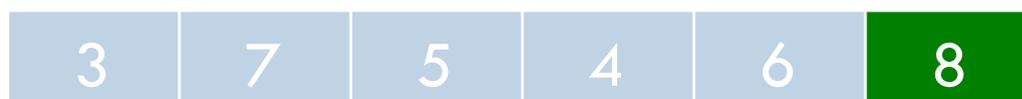
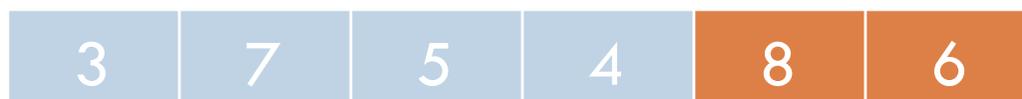
Ordinare per fusione: il MergeSort



Ordinamento a bolle: il BubbleSort



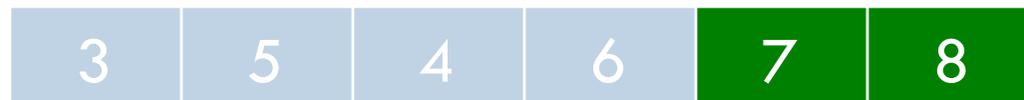
Prima passata



Ordinamento a bolle: il BubbleSort



Seconda passata



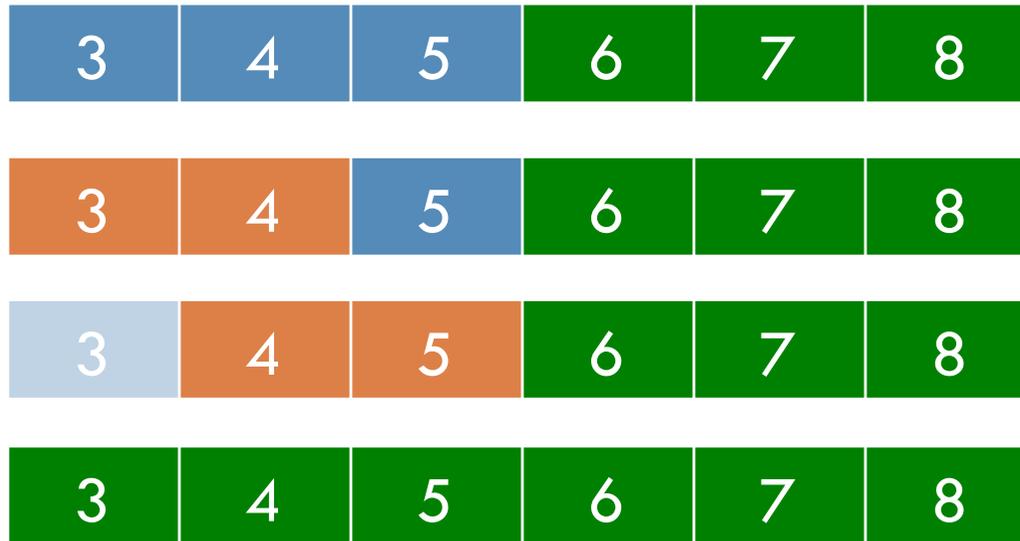
Ordinamento a bolle: il BubbleSort



Terza passata



Ordinamento a bolle: il BubbleSort



Ultima passata
(nessuno scambio)

MergeSort vs. BubbleSort



Un esperimento...

16 volontari



C'è un limite all'efficienza?

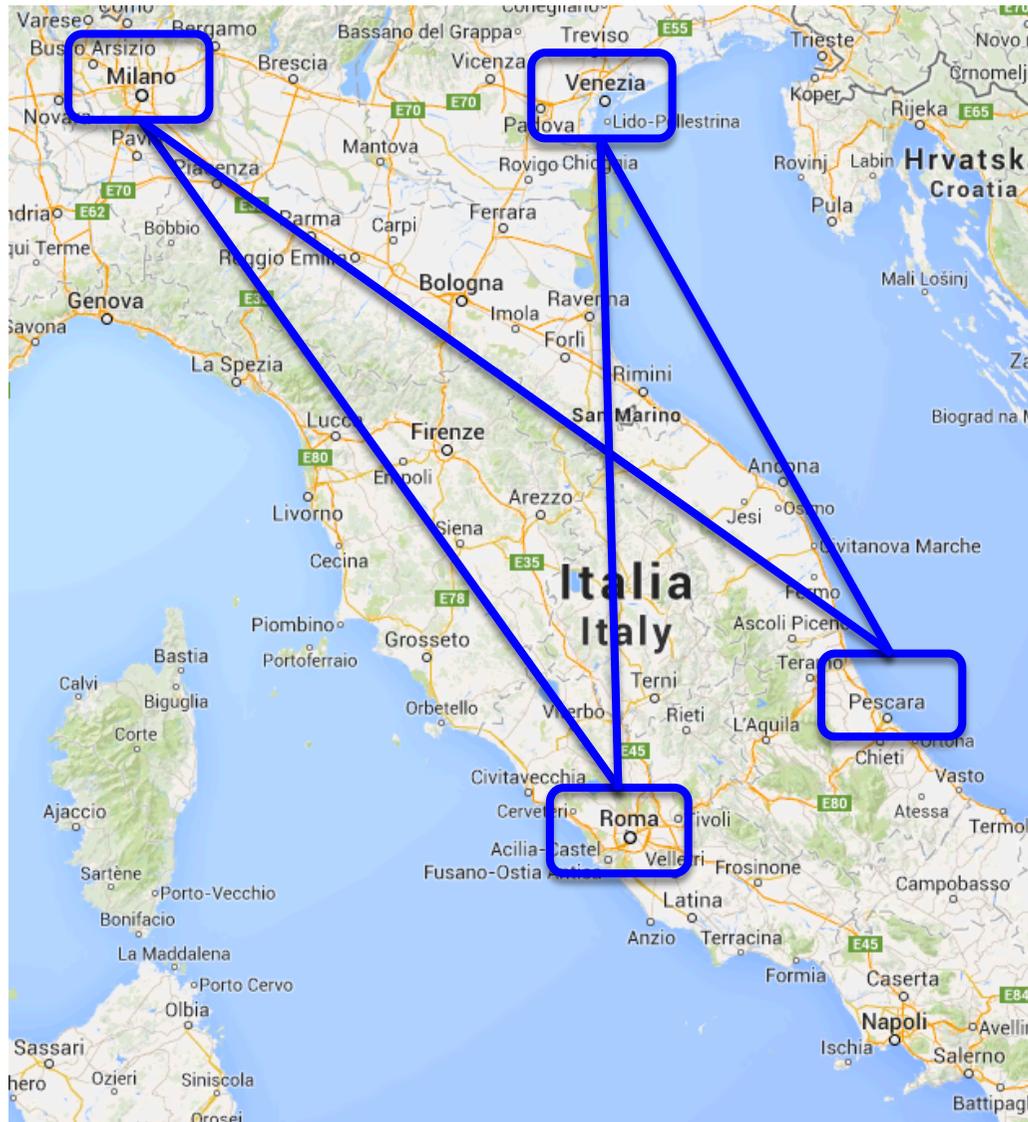
Problemi facili e problemi difficili

Il commesso viaggiatore



- Input = un insieme di città e la distanza tra ogni coppia di città
- Output = un ciclo di lunghezza minima che visita ogni città una ed una sola volta e torna alla città di partenza

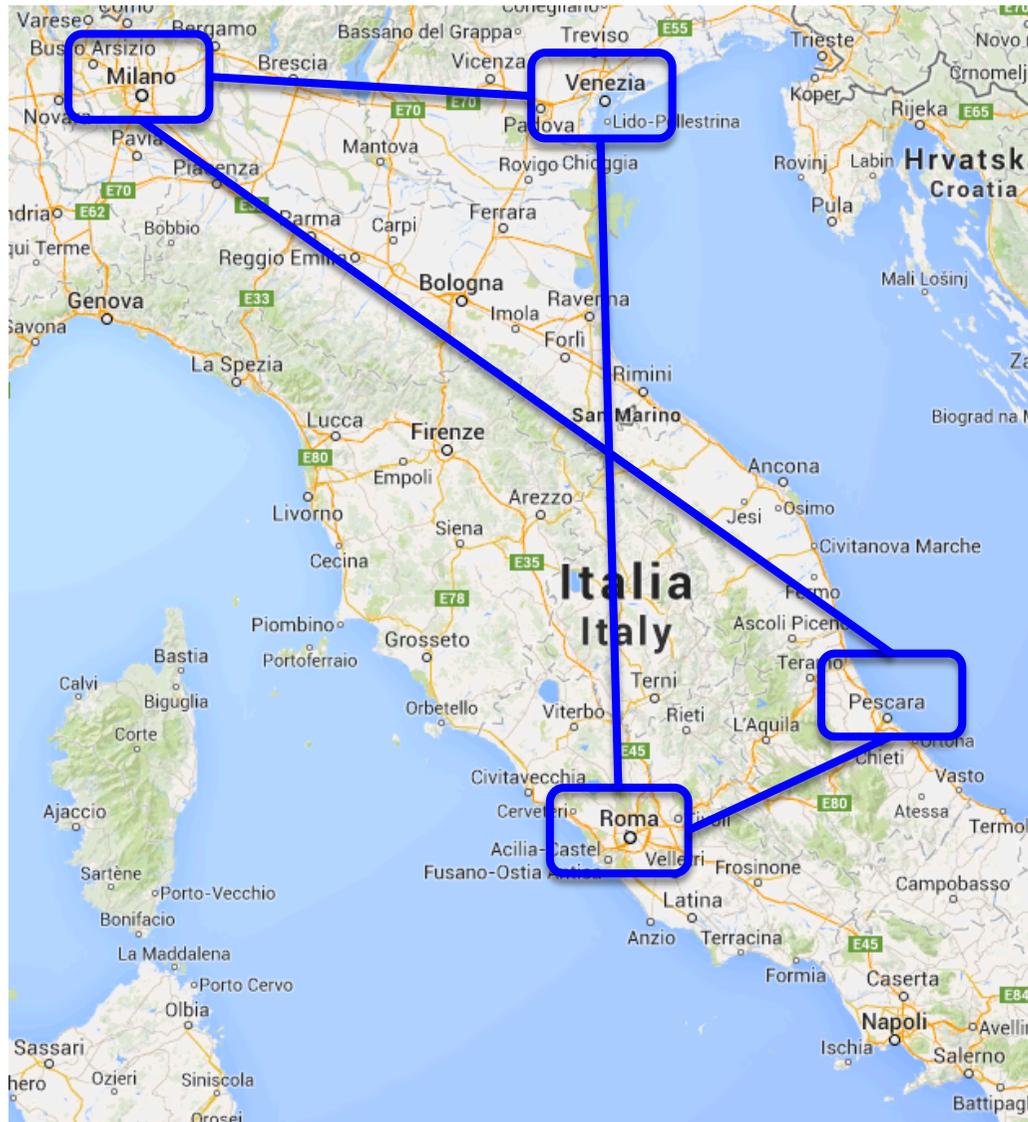
Il commesso viaggiatore



- Input = un insieme di città e la distanza tra ogni coppia di città
- Output = un ciclo di lunghezza minima che visita ogni città una ed una sola volta e torna alla città di partenza

Lunghezza = 2198 Km

Il commesso viaggiatore



- Input = un insieme di città e la distanza tra ogni coppia di città
- Output = un ciclo di lunghezza minima che visita ogni città una ed una sola volta e torna alla città di partenza

Lunghezza = 1582 Km

Il commesso viaggiatore



- Input = un insieme di città e la distanza tra ogni coppia di città
- Output = un ciclo di lunghezza minima che visita ogni città una ed una sola volta e torna alla città di partenza

Lunghezza = 1576 Km

Esiste un algoritmo efficiente per TSP?



- Efficiente = polinomiale
- Si congetture che non esista nessun algoritmo che in tempo polinomiale possa calcolare il tour migliore
- Moltissimi problemi di questo tipo!



Spunti di riflessione

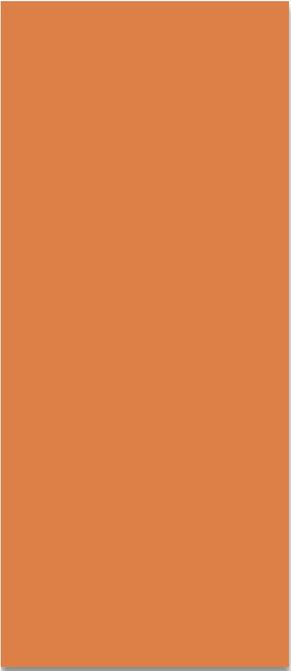
Libri divulgativi e ricreativi



- Giorgio Ausiello & Rossella Petreschi eds., *L'informatica invisibile: come gli algoritmi regolano la nostra vita... e tutto il resto*, Mondadori, 2010
- David Harel & Yishai Feldman, *Algorithmics: the spirit of computing*, 3rd ed., Springer, 2012
- Fabrizio Luccio & Linda Pagli, *Algoritmi, divinità e gente comune*, Edizioni ETS, 1999
- Dennis E. Shasha, *Puzzles for programmers and pros*, Wrox-Wiley, 2007

... e alcuni siti Web

- CS4FN: Computer Science for Fun
<http://www.cs4fn.org/>
- CS Unplugged: Computer Science... without a computer
<http://csunplugged.org/>
- Algoritmi di ordinamento (bubblesort 0:09, mergesort 0:54)
<https://www.youtube.com/watch?v=t8g-iYGHpEA>
- Bubblesort and mergesort dance
<https://www.youtube.com/watch?v=lv3vgjM8Pv4>
https://www.youtube.com/watch?v=XaqR3G_NVoo



L'informatica non riguarda i computer
più di quanto l'astronomia riguardi i
telescopi

Edsger W. Dijkstra