Ulteriori note sugli array

Eugenio G. Omodeo

UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Trieste, 27/10/2015

Cosa s'intende per *sorting* ?

Definizione (Da Wikipedia):

Sorting is any process of arranging items according to a certain sequence or in different sets, and therefore, it has two common, yet distinct meanings:

1. ordering: arranging items of the same kind, class or nature, in some ordered sequence,
2. categorizing: grouping and labeling items with similar

Cosa s'intende per *sorting* ?

Ha sempre senso disporre le componenti di un *array* in ordine non-decrescente (o al contrario)? Occorre che:

1

Cosa s'intende per *sorting* ?

Ha sempre senso disporre le componenti di un *array* in ordine non-decrescente (o al contrario)? Occorre che:

1. le componenti appartengano a un dominio dove si possono effettuare confronti

---

1

Cosa s'intende per *sorting* ?

Ha sempre senso disporre le componenti di un *array* in ordine non-decrescente (o al contrario)? Occorre che:

1. le componenti appartengano a un dominio dove si possono effettuare confronti
2. l'*array* sia utilizzato per rappresentare un *insieme* o un *multi*-insieme[1]

[1] Nel secondo caso, eventuali ripetizioni vengono considerate rilevanti.

Dov'è l'aspetto pratico ?

Dov'è l'aspetto pratico ?

Pensate alla ricerca delle voci in un dizionario. . .

Dov'è l'aspetto pratico ?

Una buona strategia dimezza, a ogni ispezione, lo spazio di ricerca

. . . UN IMPORTANTE PRINCIPIO . . .

Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information

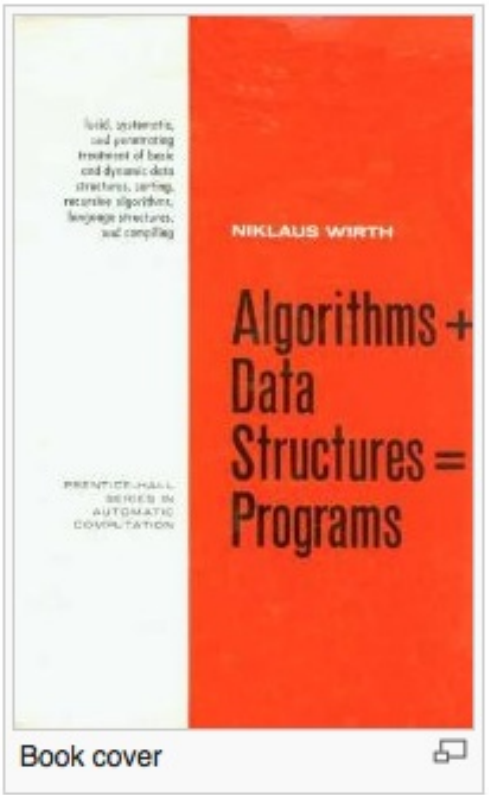
Algorithms + Data Structures = Programs

From Wikipedia, the free encyclopedia

Algorithms + Data Structures = Programs[1] is a 1976 book written by Niklaus Wirth covering some of the fundamental topics of computer programming, particularly that algorithms and data structures are inherently related. For example, if one has a sorted list one will use a search algorithm optimal for sorted lists.

The book was one of the most influential computer science books of the time and, like Wirth's other work, was extensively used in education.[2]

The Turbo Pascal compiler written by Anders Hejlsberg was largely inspired by the "Tiny Pascal" compiler in Niklaus Wirth's book.



Book cover

Chapter outline [edit]

- Chapter 1 - Fundamental Data Structures
- Chapter 2 - Sorting

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1.
2.
3.
4.

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1. Estraggo la voce ϵ che è al centro del dizionario.
2.
3.
4.

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1. Estraggo la voce ϵ che è al centro del dizionario.
2. Vale $\upsilon < \epsilon$? Escludo tutte le voci da ϵ in avanti e torno al passo 1
3.
4.

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1. Estraggo la voce ϵ che è al centro del dizionario.
2. Vale $\upsilon < \epsilon$? Escludo tutte le voci da ϵ in avanti e torno al passo 1
3. Vale $\upsilon > \epsilon$? Escludo tutte le voci da ϵ indietro e torno al passo 1
4.

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1. Estraggo la voce ϵ che è al centro del dizionario.
2. Vale $\upsilon < \epsilon$? Escludo tutte le voci da ϵ in avanti e torno al passo 1
3. Vale $\upsilon > \epsilon$? Escludo tutte le voci da ϵ indietro e torno al passo 1
4. Se sono qui ho individuato la voce: risposta affermativa

Strategia di ricerca di υ

0. Sto cercando la voce υ, inizialmente nell'intero dizionario. Progressivamente esaminerò segmenti del dizionario iniziale sempre piú piccoli e chiamerò 'dizionario' solo il segmento inesplorato.
1. Estraggo la voce ϵ che è al centro del dizionario. Non ce ne sono piú? Ho ottenuto riposta negativa
2. Vale $\upsilon < \epsilon$? Escludo tutte le voci da ϵ in avanti e torno al passo 1
3. Vale $\upsilon > \epsilon$? Escludo tutte le voci da ϵ indietro e torno al passo 1
4. Se sono qui ho individuato la voce: risposta affermativa

Firma del metodo che stiamo per vedere:

```
public static boolean biCerca(int voce, int[] dizionario)

    // Ricerca tramite metodo di bisezione
```

Firma del metodo che stiamo per vedere:

public static boolean biCerca(int voce, int[] dizionario)

// Ricerca tramite metodo di bisezione

Dizionario online tratto da:

Grande Dizionario Italiano
di **GABRIELLI ALDO**
Dizionario della Lingua Italiana

Editore: **HOEPLI**

CERCA IL SIGNIFICATO

CERCA

» bisessuale
» bisessualità
» bisessuato
» bisestile
» bisesto

bisezione

STAMPA

[bi-se-zió-ne]
s.f. (pl. *-ni*)
GEOM Divisione di un angolo piano o di un diedro in due parti uguali

Firma del metodo che stiamo per vedere:

```
public static boolean biCerca(int voce, int[] dizionario)

// Ricerca tramite metodo di bisezione
```

Dizionario online tratto da:

Grande Dizionario Italiano
di **GABRIELLI ALDO**
Dizionario della Lingua Italiana

Editore: **HOEPLI**

CERCA IL SIGNIFICATO

CERCA

- bisessuale
- bisessualità
- bisessuato
- bisestile

bisezione

STAMPA

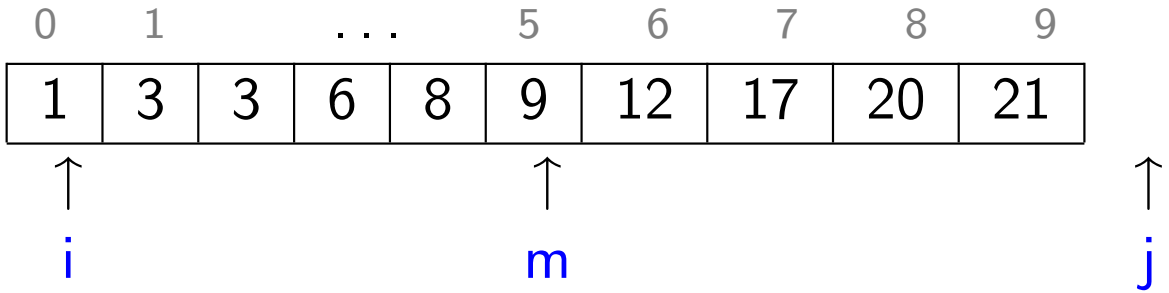
[bi-se-zió-ne]
s.f. (pl. *-ni*)
GEOM Divisione di un angolo piano o di un diedro in due parti uguali

(Implementeremo un rozzo prototipo !)

Ricerca per bisezione, in Java

```
public static boolean biCerca_iter( int voce, int[] elenco ) {

// cerca 'voce' entro elenco ordinato
// implementazione iterativa

   int i = 0, j = elenco.length, m;

   while( i < j ) { // finche` c'e` spazio entro cui cercare

      m = i + (j - i) / 2 ; // posizione di mezzo fra i e j

      if ( voce > elenco[ m ] ) i = m + 1; // guarda a dx

      else if ( voce < elenco[ m ]) j = m; // guarda a sn

      else return true;
   }

   return false;
}
```

Il metodo di ricerca in azione




```
class Bicerca{

//    | 1| 3| 3| 6| 8| 9|12|17|20|21|    ORDINE NON DECRESCENTE
//     -- -- -- -- -- -- -- -- -- --
//      i              m             j   CERCO 15
//
//                        i     m    j
//                           m  j
//                           j

    public static void main( String[] aa ){

        int[ ] diz = { 1, 3, 3, 6, 8, 9, 12, 17, 20, 21 };

        System.out.println( biCerca_iter(15, diz) );
    }
```

L'algoritmo di *sorting* più semplice di tutti

Sia a l'*array* dato:

0. Partire con $\ell = a.length - 1$;
1. se $\ell \leqslant 0$ restituire il risultato ;
2. individuare la posizione p del (o di un) massimo fra

$$a[0], \ldots, a[\ell] \,;$$

3. scambiare $a[p]$ con $a[\ell]$;
4. decrementare ℓ di un'unità e tornare al passo 1.

Scambio: ingrediente-base per metodi di *sorting*

```
class Swap{

// Metodo per lo scambio le componenti nelle posizioni indicate di un array.

// Che l'array sia formato da interi e` del tutto irrilevante.

    private static void swap( int[ ] a, int p, int q ) {

        int t = a[ q ]; // salvataggio temporaneo

        a[q] = a[ p ];

        a[p] = t ;
    }
```

Scambio: ingrediente-base per metodi di *sorting*

```
// Sempre in tema di swap:

// Esercizio: Sapreste effettuare una rotazione `sul posto'
//  di tutte le componenti dell'array
// (di un numero di posizioni indicato da un parametro intero,
// positivo o negativo a seconda che si voglia un senso di
// rotazione orario o antiorario)?
//
// Nota bene: L'elemento che esce da una parte deve rientrare
// dall'altra.

// Esercizio: Sapreste fare lo stesso con una cornice di un
// array bidimensionale quadrato?
```

Sia a l'*array* dato:

0. Partire con $\ell = a.length - 1$;
1. se $\ell \leqslant 0$ restituire il risultato ;
2. percorrere nell'ordine da sn a dx gli elementi

$$a[1], \ldots, a[\ell]$$

   o ogniqualvolta risulta che $a[i] > a[i-1]$, effettuare lo scambio fra i due;
3. decrementare ℓ di un'unità e tornare al passo 1.

Miglioramento dell'algoritmo *bubblesort*

Se nel corso di una 'spazzata' non avviene nessuno scambio, concludere. Ulteriori spazzate—piú brevi—avrebbero effetto nullo.

Esercizi

1. implementare in Java i 3 metodi di sorting descritti sopra (o, quanto meno, il *bubble-sort* migliorato;

Esercizi

1. implementare in Java i 3 metodi di sorting descritti sopra (o, quanto meno, il *bubble-sort* migliorato;
2. fatto ciò, reimplementare la `class` `Swap` utilizzando il metodo della tripla `xor`