

Algoritmi

Paolo Bison

Fondamenti di Informatica

A.A. 2001/02

October 11, 2001

- un insieme di passi/istruzioni che definiscono una sequenza di operazioni mediante le quali si risolve un problema (o una classe di problemi)
- proprietà

finitezza

numero dei passi finito ed eseguito un numero finito di volte

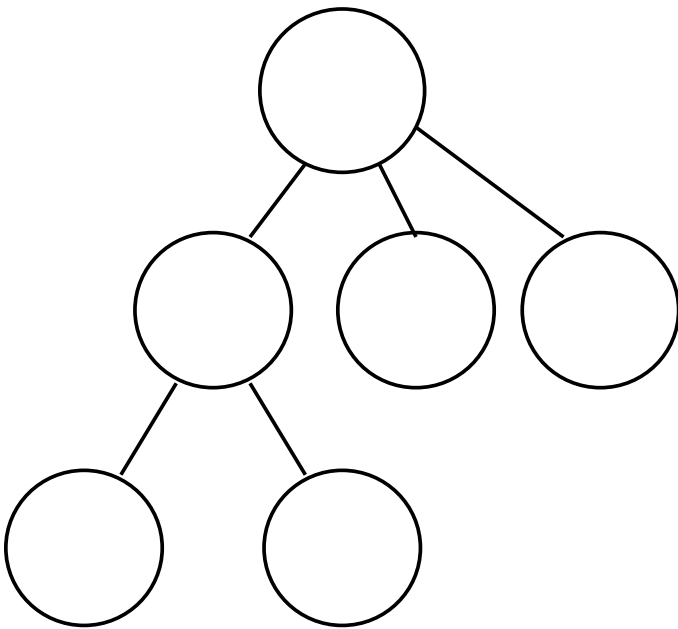
determinatezza

istruzioni definite senza ambiguità in modo tale da garantire sempre gli stessi risultati a parità di condizioni

realizzabilità

istruzioni devono essere effettivamente realizzabili, cioè conosciute all'esecutore che sa come eseguirle

- approccio metodologico alla progettazione di algoritmi per affinamenti successivi
- divide et impera
- fasi
 - dato un problema si scompone in sottoproblemi
 - ciascun sottoproblema è a sua volta scomposto in ulteriori sottoproblemi
 - così via finché ogni sottoproblema è sufficientemente dettagliato e preciso da poter essere eseguito.



Algoritmo del tè

- algoritmo principale:
 - fai bollire dell'acqua
 - metti il tè nella tazza
 - versa l'acqua
- da notare:
 - azioni(istruzioni) e oggetti (dati)
 - ambiguità

Algoritmo del tè

- affinamento primo sottoproblema
- fai bollire l'acqua:
 - riempi il bollitore
 - accendi il fuoco
 - attendi che bolla
 - spegni il fuoco

Algoritmo del tè

- accendi fuoco:
 - 1) – usa l'accendino
 - 2) – focalizza raggi solari con una lente
 - attendi finché il fuoco appare
 - 3) – strofina due legnetti
 - continua finché il fuoco appare
 - 4) – attendi che un fulmine colpisca un albero
 - usa un ramo ardente
- scelta dipende dalle risorse

Bottom Up

- approccio opposto al top down
- dato un certo insieme di algoritmi/istruzioni elementari si cerca di aggregarli in maniera da costruire l'algoritmo per la soluzione del problema o di suoi sotto-problemi
- combinazione delle due metodologie

Rappresentazione

- linguaggi per rappresentare algoritmi:
 - naturale
 - pseudocodice
 - flow chart
 - linguaggio di programmazione

Pseudocodice

- linguaggio testuale
- mix di linguaggio naturale ed elementi linguistici la cui sintassi e semantica sono ben definite ed univoche
- elementi base
 - struttura sequenziale
 - struttura di selezione
 - strutture iterative
 - variabile ed istruzione di assegnazione

Sequenziale

- sequenza di passi da eseguirsi una di seguito all'altra
- semantica
 - passi eseguiti uno alla volta
 - ciascun passo è eseguito una sola volta e nessuno è omesso o ripetuto
 - l'ordine di esecuzione è quello di scrittura
 - algoritmo termina con il termine dell'ultimo passo
- struttura rigida
- esecuzione fissa e non può essere modificata dalle circostanze

Che fare se si vuole limone o latte e/o zucchero nel tè?

- permette di eseguire istruzioni differenti al verificarsi o meno di una condizione
- sintassi
 - if** condizione
 - then** istr_1
 - else** istr_2
- semantica
 - se la condizione è vera
 - si esegue istr_1
 - altrimenti istr_2
- una sola via
 - 1) **if** condizione
 - then** istr_1
 - else**
 - 2) **if** condizione
 - then** istr_1
- **if** vuoi tè verde
- then** prendi la scatola verde
- else** prendi la scatola rossa
- esecutore deve essere in grado di interpretare e valutare la condizione

Gerarchie di selezione

- sequenze in cascata di costrutti di selezione
- scelta del massimo tra tre numeri X,Y e Z

```
if X > Y
  then
    if X > Z
      then X è max
      else Z è max
    else
      if Y > Z
        then Y è max
        else Z è max
```

- numero di vie selezionabili arbitrario ma finito
- indentazione

- ricerca di un indirizzo in un archivio dato il nome

leggi nome della prima scheda

if è il nome cercato

then estrai indirizzo

else

leggi nome della seconda scheda

if è il nome cercato

then estrai indirizzo

else

if

- non è possibile esprimere algoritmi la cui lunghezza dipenda da fattori esterni
- strutture iterative
ripetizione di istruzioni per un numero arbitrario finito di volte
- ciclo (loop)
- permettono di descrivere una elaborazione di durata indeterminata con un numero finito di istruzioni

- sintassi

repeat

istr

until condizione

- semantica

- si esegue istr
- si valuta la condizione
- se falsa si riesegue istr
altrimenti termina l'esecuzione

- ricerca in un archivio

leggi primo nome

repeat

if è il nome cercato

then estrai indirizzo

else

leggi nome successivo

until trovi il nome o elenco esaurito

- condizione di terminazione

deve assicurare che l'iterazione termini
correttamente

- ciclo errato se archivio vuoto

- sintassi

while condizione **do**

istr

- semantica

- si valuta la condizione

- se vera

- si esegue istr

- si rivaluta la condizione

altrimenti termina l'esecuzione

- ricerca del max in un elenco di numeri

stabilisci come max il primo numero

while elenco non è esaurito **do**

esamina numero successivo

if questo > max finora trovato

then stabilisci questo come max

6 -1 7 4 10

- quando si conosce il numero di iterazioni prima dell'esecuzione del ciclo
- sintassi
 - repeat N times**
istr
- semantica
 - si esegue N volte istr
- quanto zucchero?
 - chiedi quanto zucchero
 - repeat** il numero dato in risposta **times**
metti zucchero nella tazza
- iterazione definita
 - durata determinata
 - termine garantito
- iterazione indefinita
 - durata indeterminata
 - termine dipende dal verificarsi o meno della condizione di terminazione

variabile

- elemento di un algoritmo che può assumere qualunque valore preso da un dato insieme e che in ogni momento dell'esecuzione dell'algoritmo è associato ad uno ed uno solo valore
- identificatore
sequenza di caratteri alfanumerici (nome)
es. I5 per rappresentare un numero intero
- operazioni
 - accesso al valore attuale
 stampa I5 I5+5
 - modifica del valore associato
 - istruzione di assegnazione
 - operazioni di lettura
 leggi I5

istruzione di assegnazione

- sintassi

$\text{id} \leftarrow \text{espressione}$

- semantica

al termine dell'esecuzione alla variabile id è associato il valore ottenuto valutando l'espressione

- esempi

$\text{l5} \leftarrow 34$

$\text{m} \leftarrow \text{l} \times \text{r}$

- ordine di esecuzione

$\begin{matrix} \text{n} \leftarrow \text{m} \\ \text{m} \leftarrow \text{r} \end{matrix} \neq \begin{matrix} \text{m} \leftarrow \text{r} \\ \text{n} \leftarrow \text{m} \end{matrix}$

- scambio di valori tra m e n

$\text{t} \leftarrow \text{m}$

$\text{m} \leftarrow \text{n}$

$\text{n} \leftarrow \text{t}$

- variabili associate ad un insieme di elementi presi da un dato insieme
- singoli elementi accessibili attraverso un indice
- notazione - data la variabile A

A l'insieme degli elementi

A_0 l'elemento di indice 0

A_k il k-esimo elemento

- operazioni

- variabile

$B \leftarrow A$ $B \leftarrow A + C$

- singolo elemento

$B_k \leftarrow A_1 - C_n$

- creazione di un array contenente i primi n numeri interi

$i \leftarrow 1$

while $i \leq n$ **do**

$A_i \leftarrow i$

$i \leftarrow i + 1$

- multidimensionali

$D_{k,j}$

somma di n numeri

calcolo della somma dei primi N numeri interi positivi

leggi n

$s \leftarrow 0$

$i \leftarrow 1$

while $i \leq n$ **do**

$s \leftarrow s + i$

$i \leftarrow i + 1$

stampa s

somma di n numeri pari

calcolo della somma dei primi N numeri interi positivi pari

leggi n

$s \leftarrow 0$

$i \leftarrow 1$

while $i \leq n$ **do**

if resto di $i/2 = 0$ **then**

$s \leftarrow s + i$

$i \leftarrow i + 1$

stampa s

resto della divisione

calcolo del resto della divisione tra due interi n e m , con $m > n$,
avendo a disposizione l'operazione di
divisione intera ($/$)

leggi n e m

$r \leftarrow m - ((m / n) \times n)$

stampa r

Fattoriale

$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1$ $n \geq 0$ e con $0! = 1$

Ciclo da n a 1 che moltiplica tutti i numeri tra n e 1

leggi n

$i \leftarrow 1$

while $n > 1$ **do**

$i \leftarrow i \times n$

$n \leftarrow n - 1$

stampa i

massimo di un array

Dati un array A di n elementi interi trovare quello massimo

leggi n e A_k ($k = 1 \dots n$)

$\max \leftarrow A_1$

$i \leftarrow 2$

while $i \leq n$ **do**

if $\max < A_i$ **then**

$\max \leftarrow A_i$

$i \leftarrow i + 1$

stampa \max

Dati due numeri $m, n > 0$ trovare MCD

Algoritmo 1

Sia $m \geq n$, con ciclo da 2 a n si verifica quali sono i numeri che dividono esattamente sia m che n . Il MCD è il massimo di tali numeri.

Nota: un numero è divisibile per un altro se il resto della divisione è zero.

leggi m, n

if $m < n$ **then** scambia m con n

$i \leftarrow 2$

$mcd \leftarrow 1$

while $i \leq n$ **do**

if resto di $m/i = 0$ **then**

if resto di $n/i = 0$ **then**

if $i > mcd$ **then**

$mcd \leftarrow i$

$i \leftarrow i + 1$

stampa mcd

Algoritmo 2 - Metodo di Euclide

Sia $m \geq n$, qualunque numero che divide sia m che n divide anche il resto della divisione m/n

$$\begin{aligned} m &= qn + r \\ m - qn &= r \geq 0 \\ q_m k - q q_n k &= r \\ k(q_m - q q_n) &= r \end{aligned}$$

Si calcola il resto r di m/n . Se tale resto è zero n è il MCD, altrimenti n e r diventano m e n e si riapplica il passo precedente.

leggi m, n

if $m < n$ **then** scambia m con n

$r \leftarrow$ resto di m/n

while $r \neq 0$ **do**

$m \leftarrow n$

$n \leftarrow r$

$r \leftarrow$ resto di m/n

stampa n

Massimo Comun Divisore

Algoritmo 3 - Metodo di Euclide (senza divisione)

Se $m=n$ il MCD è m , altrimenti se $m > n$ m diventa $m-n$ altrimenti è n che diventa $n - m$, e si ricontrolla l'eventuale uguaglianza di m con n

leggi m, n

while $m \neq n$ **do**

if $m > n$

then $m \leftarrow m - n$

else $n \leftarrow n - m$

stampa m

Numero primo

Dato un numero intero $n > 0$ si dica se è primo

Ciclo che verifichi che N non sia divisibile da un numero tra $n/2$ e 2.

leggi n

$\text{div} \leftarrow n / 2$

repeat

$r \leftarrow \text{resto di } n / \text{div}$

$\text{div} \leftarrow \text{div} - 1$

until $r = 0$ o $\text{div} = 1$

if $r \neq 0$

then stampa che n è primo

else stampa che n non è primo

Si trovino tutti i numeri primi compresi tra 1 e N

Ciclo da 1 a N di verifica se il numero è primo o no.

leggi N

$i \leftarrow 1$

repeat N times

if i è numero primo

then stampa i

$i \leftarrow i + 1$

i è numero primo

$div \leftarrow i / 2$

repeat

$r \leftarrow \text{resto di } i / div$

$div \leftarrow div - 1$

until $r = 0$ o $div = 1$

if $r \neq 0$

then vero, i è primo

else falso, i non è primo

Equazioni di II grado - I

$$ax^2 + bx + c = 0$$

leggi a,b e c

if a=0 e b=0 **then** segnala eq. degenerare
else calcola radici

calcola radici

if a=0 **then** eq. I grado
else eq. II grado

eq. I grado

$x \leftarrow -c / a$

stampa x

eq. II grado

$\Delta \leftarrow b^2 - 4ac$

if $\Delta < 0$ **then** radici immaginarie
else radici reali

Equazioni di II grado - II

$$ax^2 + bx + c = 0$$

radici immaginarie

$$z \leftarrow \sqrt{-\Delta}$$

$$\text{parte reale} \leftarrow -b / 2a$$

$$\text{parte immaginaria} \leftarrow z / 2a$$

stampa parte reale e parte immaginaria

radici reali

$$\text{if } \Delta = 0 \text{ then } x1 \leftarrow -b / 2a$$

$$x2 \leftarrow x1$$

$$\text{else } S \leftarrow \sqrt{\Delta}$$

$$x1 \leftarrow (-b + S) / 2a$$

$$x2 \leftarrow (-b - S) / 2a$$

stampa x1 e x2

valore di un polinomio

Calcolare, per un dato valore di x, il polinomio

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

leggi x,n e A_k ($k = 0 \dots n$)

pot $\leftarrow 1$

val $\leftarrow A_0$

i $\leftarrow 1$

while i \leq n **do**

 val $\leftarrow A_i \times \text{pot} + \text{val}$

 pot $\leftarrow \text{pot} \times x$

 i $\leftarrow i + 1$

stampa val

- 1 - Minimo comune multiplo di due numeri m e n
- 2 - Calcolo della radice quadrata intera di un numero $n > 0$; la radice intera è quel numero m che soddisfa le condizioni $m^2 \leq n$ e $(m + 1)^2 > n$
- 3 - Data una serie di numeri calcolarne media e varianza
- 4 - Data una serie di numeri trovarne il valore mediano m ; m è quel valore, presente nella serie, tale per cui il 50% dei numeri è $\leq m$.
- 5 - Calcolo approssimato dell'integrale definito $\int_{x_0}^{x_1} f(x)$ come area sottesa da $f(x)$ tra x_0 e x_1 .
- 6 - Calcolo dei coefficienti dell'equazione della retta $ax + by + c = 0$ dati due punti (x_0, y_0) e (x_1, y_1)
- 7 - Calcolo dei coefficienti dell'equazione della circonferenza $x^2 + y^2 + ax + by + c = 0$ dati tre punti (x_0, y_0) , (x_1, y_1) e (x_2, y_2)
- 8 - Contare le vocali contenute in una sequenza di lettere. Nota: la sequenza può essere rappresentata con array in cui ogni elemento memorizza una lettera mentre lettere costanti possono essere scritte con la notazione '*lettera*', es. $a_1 = 'c'$ è vero se l'elemento di indice 1 contiene la lettera c.
- 9 - Trovare una parola in un testo.
- 10 - Dati N colli, caratterizzati ciascuno da un peso ed un valore monetario, da trasportare con un autoveicolo con una data portata, trovare una configurazione di carico che massimizza il valore totale del carico senza superare la portata.
- 11 - Date 8 regine sistemarle in una scacchiera 8×8 in maniera tale che nessuna possa mangiare un'altra.
- 12 - Dati N oggetti, di cui $N-1$ di peso uguale, trovare quello di peso differente dicendo se il suo peso è maggiore o minore di quello degli altri utilizzando una bilancia senza pesi.