Algoritmi notevoli

Salvatore Venticinque

Università
degli Studi
della Campania
Luigi Vanvitelli

Problema:

Data l'equazione `f(x)=0`, assunto che `f(x)` sia definita e continua in un intervallo [a,b], assunto che `f(a)*f(b)<0` e che la funzione è una funzione polinomiale di ordine n:

$f(x) = a + a_1 * x^1 + a_2 * x^2 + a_{..} * x^{..} + a_n * x^n$

Calcolare almeno uno zero della funzione nell'intervallo [a,b]

Input:

- N ed i parametri a_i

Output:

- `x: f(x)=0`

Per semplicità fissiamo N=2

Metodo Risolutivo:

Suddividiamo [a,b] in due intervalli.

`I'=[a,z] I''=[z,b]`

`z=(a+b)/2`

Se f(z)=0, abbiamo trovato la soluzione.

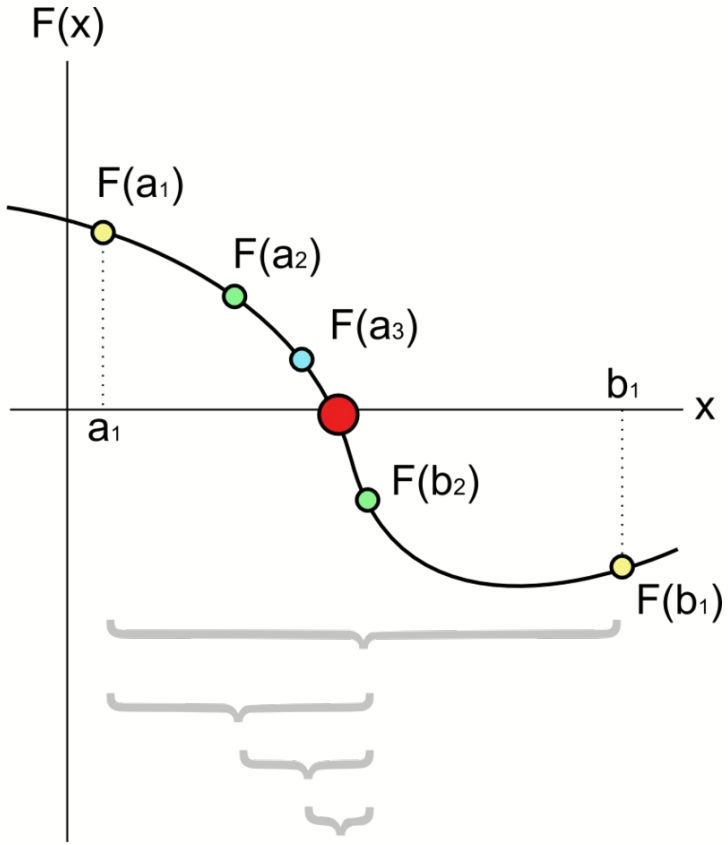
Altrimenti si sceglie

I' se `f(a)*f(z)<0`

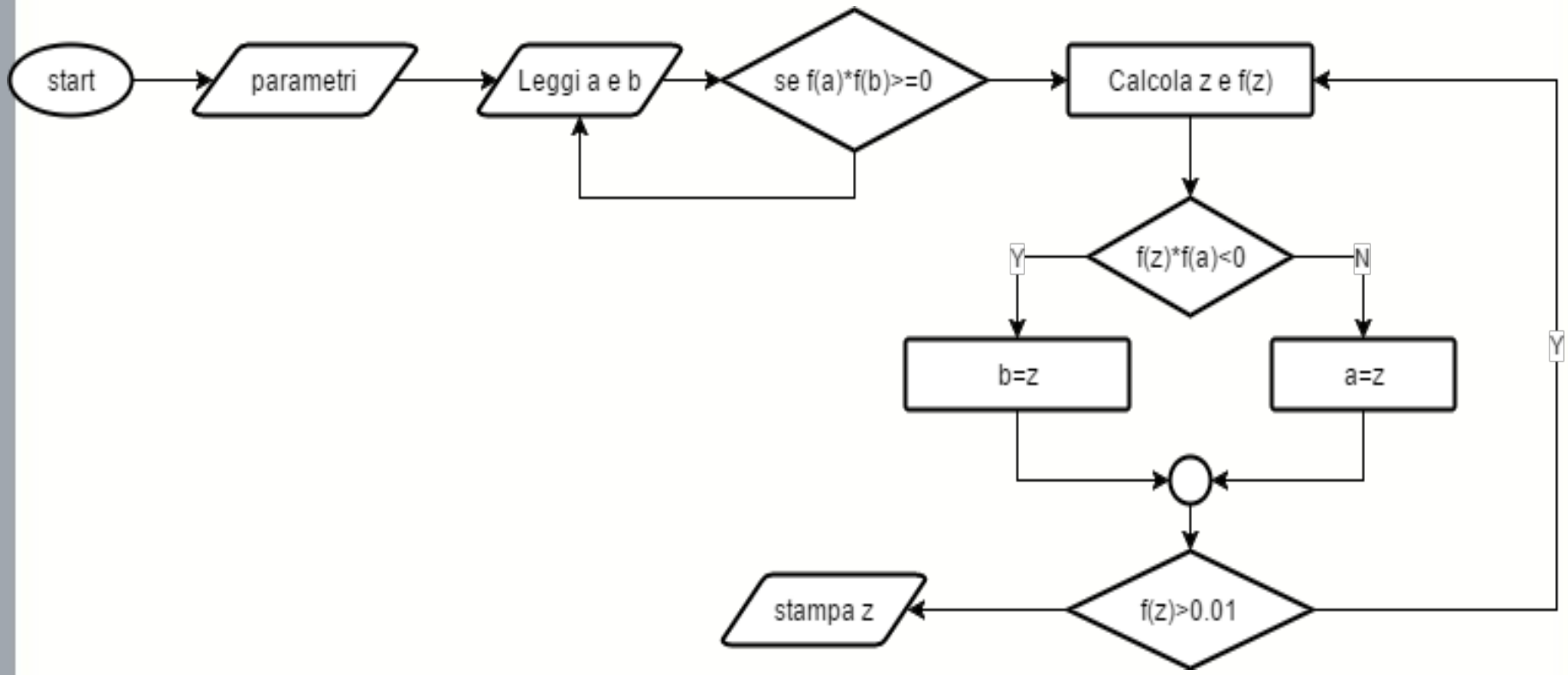
I'' se `f(z)*f(b)<0`

E si procede nuovamente.

L'algoritmo termina quando `f(z)<ε`



Zeri di una funzione
Metodo di Bisezione



Zeri di una funzione
Metodo di Bisezione

- Il Codice riporta solo la parte di elaborazione
- (Niente I/O)

```
fl=(a+a1*low+a2*low*low);
fh=(a+a1*high+a2*high*high);
do {
    z=(low+high)/2;
    fz=(a+a1*z+a2*z*z);
    if((fz*fl)<0) {
        high=z;
        fh=fz;
    } else {
        low=z;
        fl=fz;
    }
    i++;
} while((fz>0.001)&&(i<20));
```

Insertion Sort

IDEA BASE

L'algoritmo di ordinamento per inserimenti successivi si basa sulla riduzione dell'ordinamento a un problema piu semplice: l'inserimento in ordine.

Inserimento in ordine: Dato un vettore ordinato, inserire un elemento rispettando l’ordine

IDEA BASE – Inserimento in ordine

1. trova l'indice i della posizione in cui inserire x
2. sposta in avanti gli elementi di indice in modo da poter inserire x senza perdere informazioni
3. inserisci x nella posizione i dell'array

Insertion Sort
Inserimento in Ordine

```
void insert_in_order(int  a[], int n, int x) {
int pos , i;
   /* Cerca la posizione di inserimento */
   for ( pos=n; pos >0 && (a[pos -1] > x); pos--){;}
     /* Sposta in avanti gli elementi successivi */
       for (i=n-1; i >= pos ; i --)
            a[i +1]= a[i];
       a[ pos ]=x;
}
```

IDEA BASE

Prendi il primo elemento ed inseriscilo in ordine nel vettore di zero elementi, prendi il secondo elemento ed inseriscilo in ordine nel vettore di 1 elemento, etc etc..

IDEA BASE

Prendi il primo elemento ed inseriscilo in ordine nel vettore di zero elementi, prendi il secondo elemento ed inseriscilo in ordine nel vettore di 1 elemento, etc etc..

```
void insert_sort ( int V[], int n) {
    int i;
    for (i =1; i<n; i ++)
        insert_in_order (V, i, V[i]);
}
```

Merge Sort

IDEA BASE

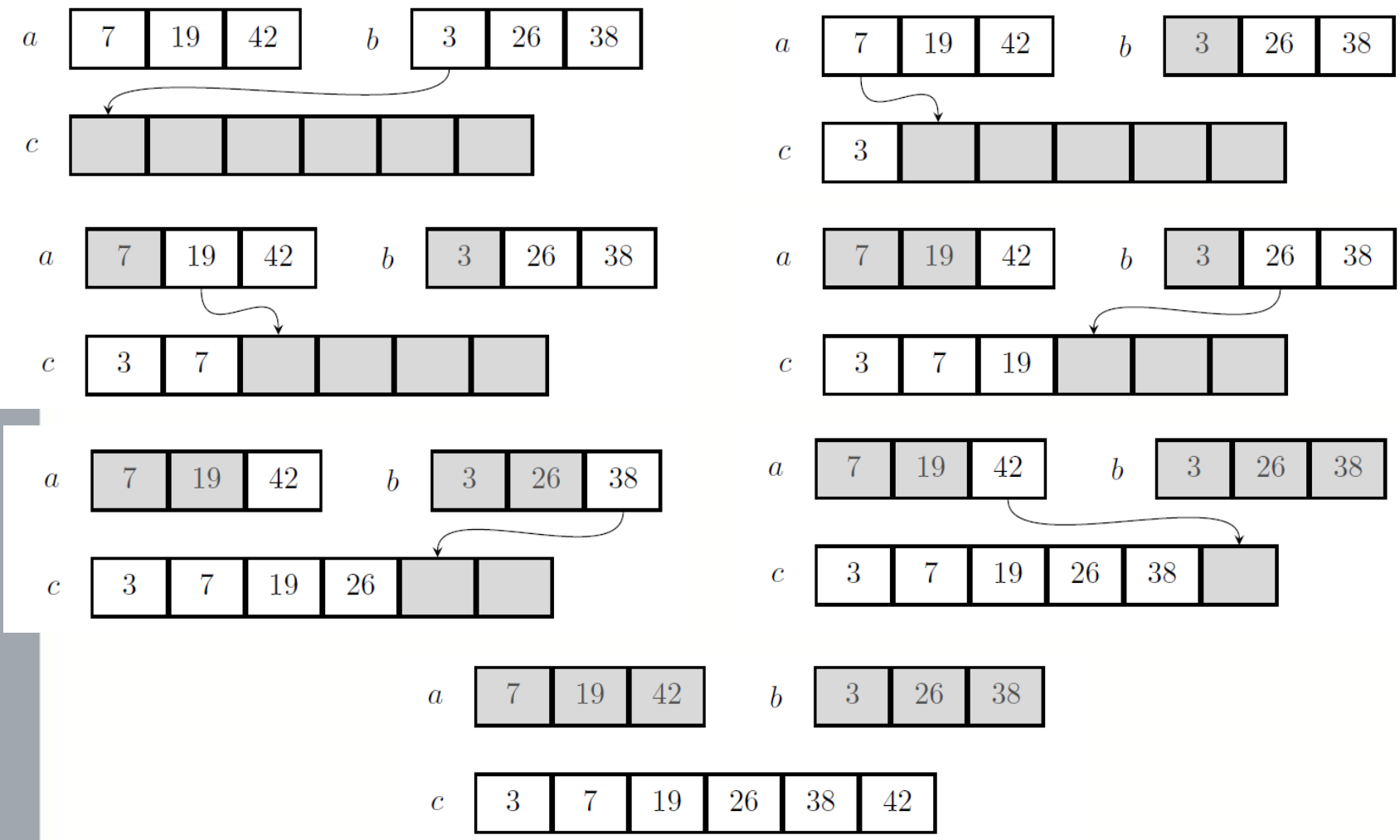
L'ordinamento per fusione o Merge Sort riconduce il problema dell'ordinamento al problema della fusione di array ordinati,

Fusione di array ordinati: Dati due array ordinati, generare un array compost dagli elementi di entrambi, anch’esso ordinato-.

IDEA BASE – Fusione ordinata

1. L'array risultato c viene costruito iterativamente, partendo da un array vuoto e aggiungendo ad ogni passo un nuovo elemento
2. Affinchè l'array c risulti ordinato possiamo aggiungere a ogni passo il più piccolo degli elementi di a e di b che non sono stati ancora usati
3. Il più piccolo tra gli elementi di a e di b è semplicemente il più piccolo tra il primo elemento di a ed il primo elemento di b non ancora inseriti

Merge Sort
Ordinamento per Fusione
- Fusione Ordinata -



Merge Sort
Ordinamento per Selezione

```
void merge ( int a1 [], int n1 , int a2 [], int n2 , int dest []) {
    int pos1 =0, pos2 =0, k =0;
    while (pos1 <n1 && pos2 <n2) {
        if ( less (a2[ pos2 ], a1[ pos1 ]))
            dest [k ++] = a2[ pos2 ++];
    else
            dest [k ++] = a1[ pos1 ++];
    }
    while (pos1 <n1)
            dest [k ++] = a1[ pos1 ++];
    while (pos2 <n2)
            dest [k ++] = a2[ pos2 ++];
}
```

- Caso base: se n= 1 allora l'array è ordinato
- **Divide**: dividiamo a in due parti, a’ e a’’ rispettivamente di m = n/2 elementi e di n-m;
- **Impera**: Applica l’algoritmo ad a’ ed a’’
- **Combina**: utilizzando l'algoritmo di fusione, fondiamo gli array ordinati a’ e a’’ producendo un nuovo array ordinato

IDEA BASE

Caso base: se n= 1 allora l'array è ordinato

Divide: dividiamo a in due parti, a’ e a’’ rispettivamente di m = n/2 elementi e di n-m;

Impera: Applica l’algoritmo ad a’ ed a’’

Combina: utilizzando l'algoritmo di fusione, fondiamo gli array ordinati a’ e a’’ producendo un nuovo array ordinato

Merge Sort
Ordinamento per Selezione

```
void merge_sort ( int a[], int n, int temp []) {
    int i, m=n/2;
    if (n <2)
        return ;
    merge_sort (a, m, temp );
    merge_sort (a+m, n-m, temp );
    merge (a, m, a+m, n-m, temp );
    for (i =0; i<n; i ++)
            a[i]= temp [i];
}
```