Gli algoritmi

Rappresentare una procedura rigorosa e generale di soluzione di un problema

Si dice **algoritmo** *un insieme di istruzioni che definiscono una sequenza di operazioni mediante le quali si risolvono tutti i problemi di una determinata classe.*

L'algoritmo deve essere:

•**finito**, costituito cioè da un numero limitato di passi (le istruzioni sono in numero finito e vengono eseguite un numero finito di volte);

•**definito**, ogni istruzione deve consentire un'interpretazione univoca;

•**eseguibile**, cioè la sua esecuzione deve essere possibile con gli strumenti di cui si dispone;

•**deterministico**, ad ogni passo deve essere definita una ed una sola operazione successiva.

Le caratteristiche sopraelencate caratterizzano un processo automatizzabile, che esclude la componente intuitiva propria dell'elaborazione governata dall'uomo. Le istruzioni devono essere espresse informa comprensibile all'esecutore (cioè interpretabili dal sistema di elaborazione) e devono descrivere le azioni da compiere al livello di dettaglio delle capacità elementari dell'esecutore.

Per facilitare la stesura dell'algoritmo si ricorre a rappresentazioni grafiche e formalizzate, quali ad esempio i **Diagrammi a Blocchi** o **il Linguaggio di progetto.**

Un primo metodo di rappresentazione della procedura di risoluzione di un problema

Tra le tecniche utilizzate per rappresentare in maniera chiara e sintetica la struttura degli algoritmi, quella del Diagramma a Blocchi , anche detto **flow-chart**, ha il pregio di evidenziare visivamente l'avanzamento in sequenza e le varie *strutture* che compongono l'algoritmo.

Questa tecnica si basa sull'osservazione che le istruzioni di un qualunque algoritmo sono di uno dei seguenti tipi:

- **istruzioni di calcolo o di elaborazione,** che descrivono un'azione da compiere necessariamente dopo la precedente e prima della seguente nell'elenco ordinato;

- **condizioni** che pongono una domanda cui si può rispondere sì/no oppure vero/falso e che a seconda della risposta indicano due diverse azioni da

compiere.

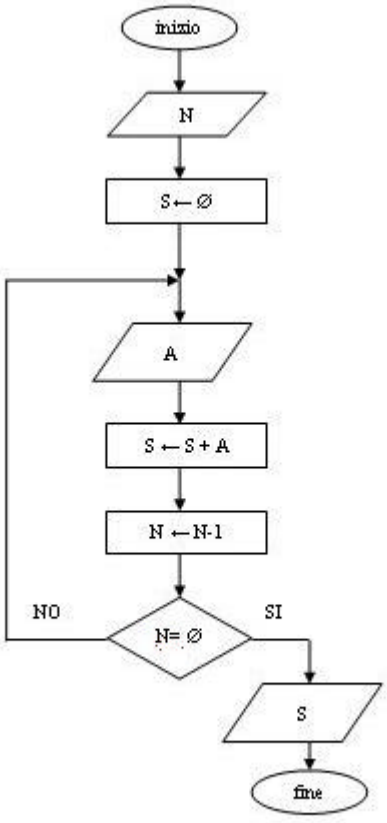
Le due tipologie di operazioni vengono rappresentate convenzionalmente con le forme geometriche rettangolo e rombo, la logica di sequenza viene data da frecce che congiungono i vari blocchi nel senso di avanzamento dell'esecuzione. Per le istruzioni di input e di output si usa un parallelogramma, per identificare l'inizio e la fine della sequenza di istruzioni si usa l'ellisse, la necessità di mutare la sequenza d'esecuzione passando ad un'istruzione che non sia quella immediatamente successiva (istruzioni di salto) è resa graficamente mediante le frecce.

Per meglio chiarire l'uso di questa tecnica di rappresentazione si vedano gli **esempi** di diagrammi a blocchi.

ALGORITMO 1 - Somma di una sequenza di numeri

Indicando con ai il generico elemento da sommare, la formula matematica generale è $S = a_1 + a_2 + \ldots + a_n$. Bisogna fornire in input all'elaboratore i singoli valori ai ed il numero n di tali valori. L'algoritmo utilizza una *struttura iterativa*, ossia un blocco d'istruzioni viene ripetuto un numero finito di volte. Utilizzeremo una variabile N per l'input di n e per contare quante volte si deve ripetere l'iterazione; il valore di N si decrementa di un'unità nell'ambito dell'iterazione e l'iterazione termina quando N raggiunge il valore zero. La variabile A è usata per gli input degli ai, S per le somme parziali e totale.

Fig. 3 *Algoritmo 1 – Diagramma a blocchi*



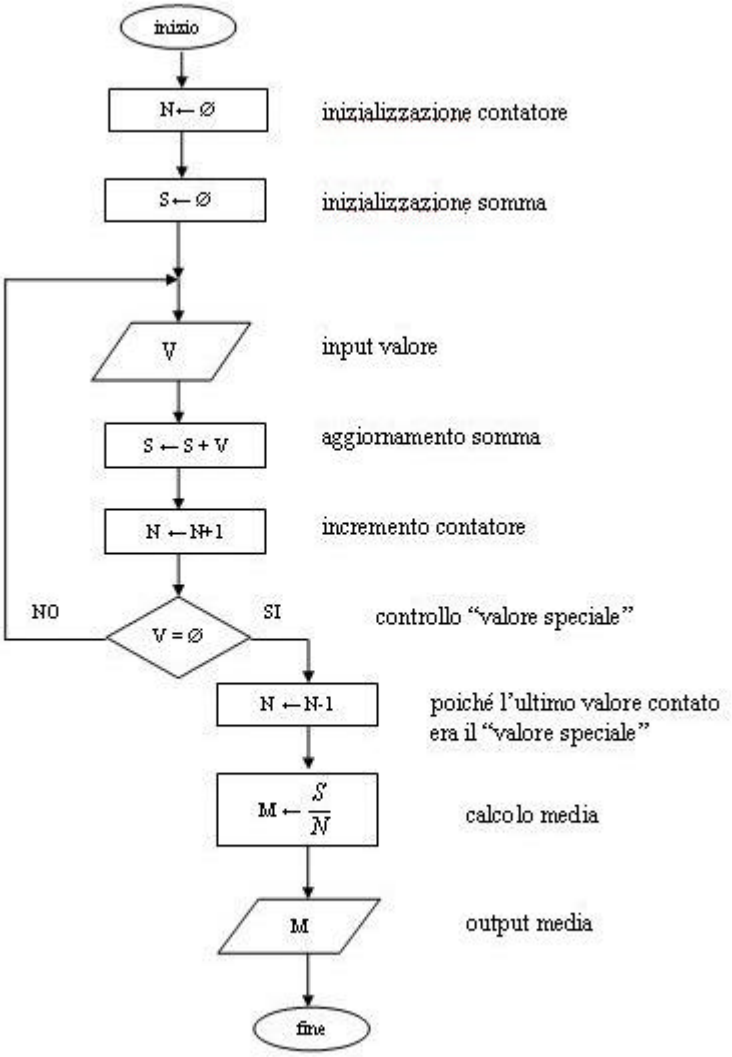
ALGORITMO 2 – Calcolo della media aritmetica di una sequenza di valori numerici

La formula matematica è

$$M = \frac{a_1 + a_2 + \ldots + a_n}{n}$$

Si consente all'utente di introdurre un numero qualsiasi di dati, utilizzando un valore speciale (lo zero) per indicare la fine della sequenza di input; l'algoritmo conta quanti elementi vengono introdotti.

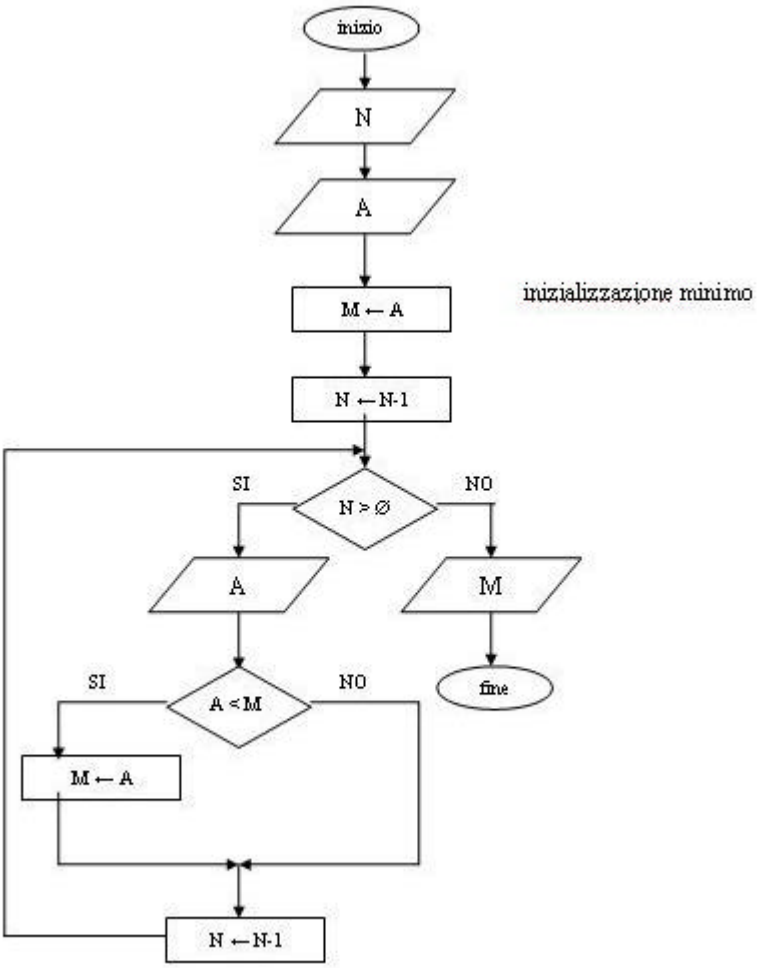
Fig.4 *Algoritmo 2 – Diagramma a blocchi*



ALGORITMO 3 – Ricerca del minimo di una sequenza di valori numerici

Indicati con ai i valori sui quali si deve effettuare la ricerca, si può risolvere il problema con un algoritmo iterativo che effettua confronti successivi: $M_1 = \min(a_1, a_2)$; $M_2 = \min(M_1, a_3)$; $M_3 = \min(M_2, a_4)$;…; $M_{n-1} = \min(M_{n-2}, a_n)$. Al termine degli n-1 confronti, M_{n-1} coincide col minimo cercato. La variabile A è usata per gli input degli a_i . N è il contatore delle iterazioni.

Fig. 5 *Algoritmo 3 – Diagramma a blocchi*

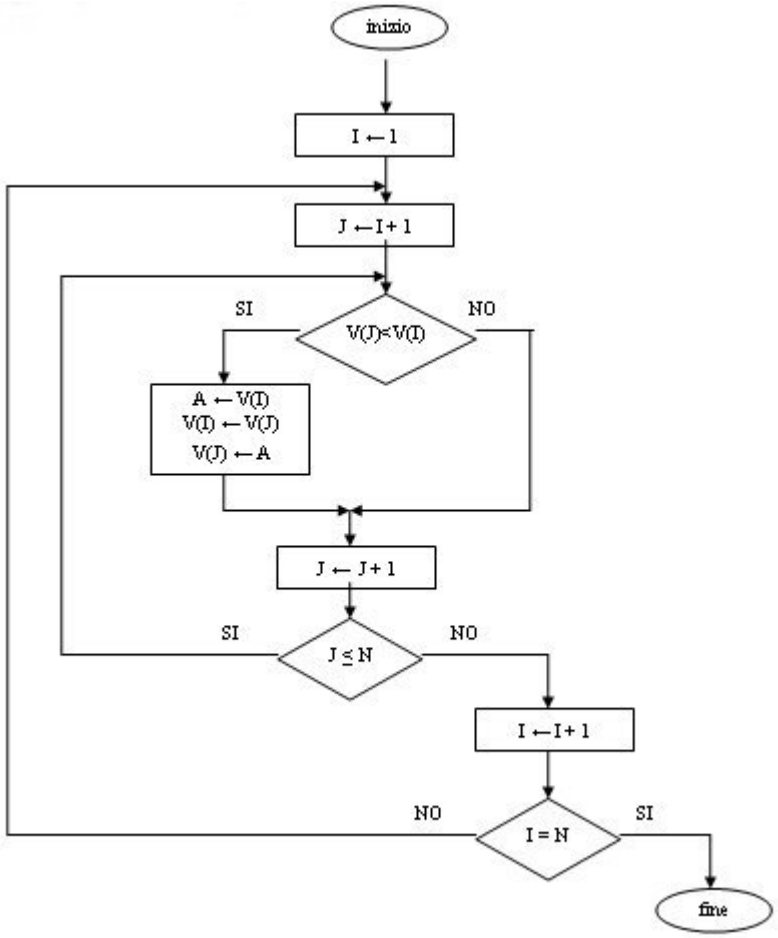


Esercizio proposto: in modo analogo, realizzare il diagramma per l'algoritmo di ricerca del massimo di una sequenza di valori.

ALGORITMO 4 – Ordinamento per scambio di una sequenza di numeri in modo crescente

Indicati con ai i valori da ordinare, deve risultare $a_1 < a_2 < a_3 < \ldots < a_{n-1} < a_n$. Si può sfruttare l'algoritmo per la ricerca del minimo, utilizzando un vettore per memorizzare gli n valori numerici da ordinare. Si inizierà applicando l'algoritmo di ricerca del minimo su tutti gli elementi del vettore e spostando il minimo in prima posizione, si procederà analogamente sui rimanenti n-1 elementi e il minimo tra essi si sposterà in seconda posizione, poi si ripeterà la procedura sui restanti n-2 elementi del vettore e così via. Il diagramma a blocchi illustra l'algoritmo in questione, supponendo di avere già a disposizione il vettore V(i) con i suoi n valori.

Fig. 6 *Algoritmo 4 – Diagramma a blocchi*



Esercizio proposto: in modo analogo, realizzare il diagramma per l'algoritmo di ordinamento per scambio di una sequenza di numeri in modo decrescente.

Un secondo metodo di rappresentazione della procedura di risoluzione di un problema

Un algoritmo può essere rappresentato anche mediante un linguaggio speciale che descrive le istruzioni e la logica di avanzamento dell'esecuzione con frasi anziché con un diagramma. Si parla in tal caso di **pseudocodifica** o **notazione lineare strutturata**.

Tale linguaggio formale, detto **linguaggio di progetto**, assomiglia al linguaggio naturale, ma è privo di ambiguità e segue regole prive di eccezioni. Esso utilizza *parole riservate,* spesso tratte dalla lingua inglese, mediante le quali vengono espressi i diversi tipi di istruzioni.

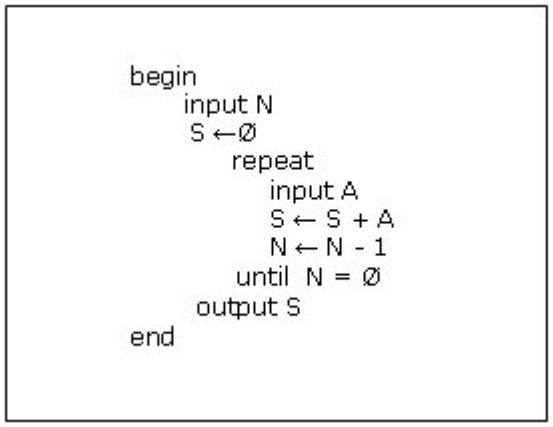
Gli **esempi** mostrano l'utilizzo di questa tecnica, in particolare mediante le cosifica in linguaggio pseudopascal, cosiddetto perché simile al linguaggio di programmazione Pascal.

Esempi

ALGORITMO 1 - Somma di una sequenza di numeri

Riguardo i dettagli dell'algoritmo si rimanda a quanto detto nella sua codifica mediante diagramma a blocchi.

Pseudocodifica algoritmo 1:



ALGORITMO 2 – Calcolo della media aritmetica di una sequenza di valori numerici

Riguardo i dettagli dell'algoritmo si rimanda a quanto detto nella sua codifica mediante diagramma a blocchi.

Pseudocodifica algoritmo 2:

```
begin
   N ← Ø
   S ← Ø
    repeat
    input V
    S ← S + V
    N ←  N + 1
    until V = Ø
  N ←  N – 1
   M ←  S/N
   output M
end
```

ALGORITMO 3 – Ricerca del minimo di una sequenza di valori numerici

Riguardo i dettagli dell'algoritmo si rimanda a quanto detto nella sua codifica mediante diagramma a blocchi.

Pseudocodifica algoritmo 3:

```
begin
    input N
    input A
     M ← A
     N ← N - 1
        while N > Ø do
          begin
            input A
          if A < M then M ← A
          N ← N - 1
        end
      output M
  end
```

ALGORITMO 4 – Ordinamento per scambio di una sequenza di numeri in modo crescente

Riguardo i dettagli dell'algoritmo si rimanda a quanto detto nella sua codifica mediante diagramma a blocchi.

Pseudocodifica algoritmo 4:

```
begin
    I ← 1
    repeat
       J ← I + 1
       repeat
          if V(J)<V(I) then
              begin
                 A ← V(I)
                 V(I) ← V(J)
                 V(J) ← A
              end
          J ← J + 1
       until J > N
       I ← I + 1
    until I = N
end
```