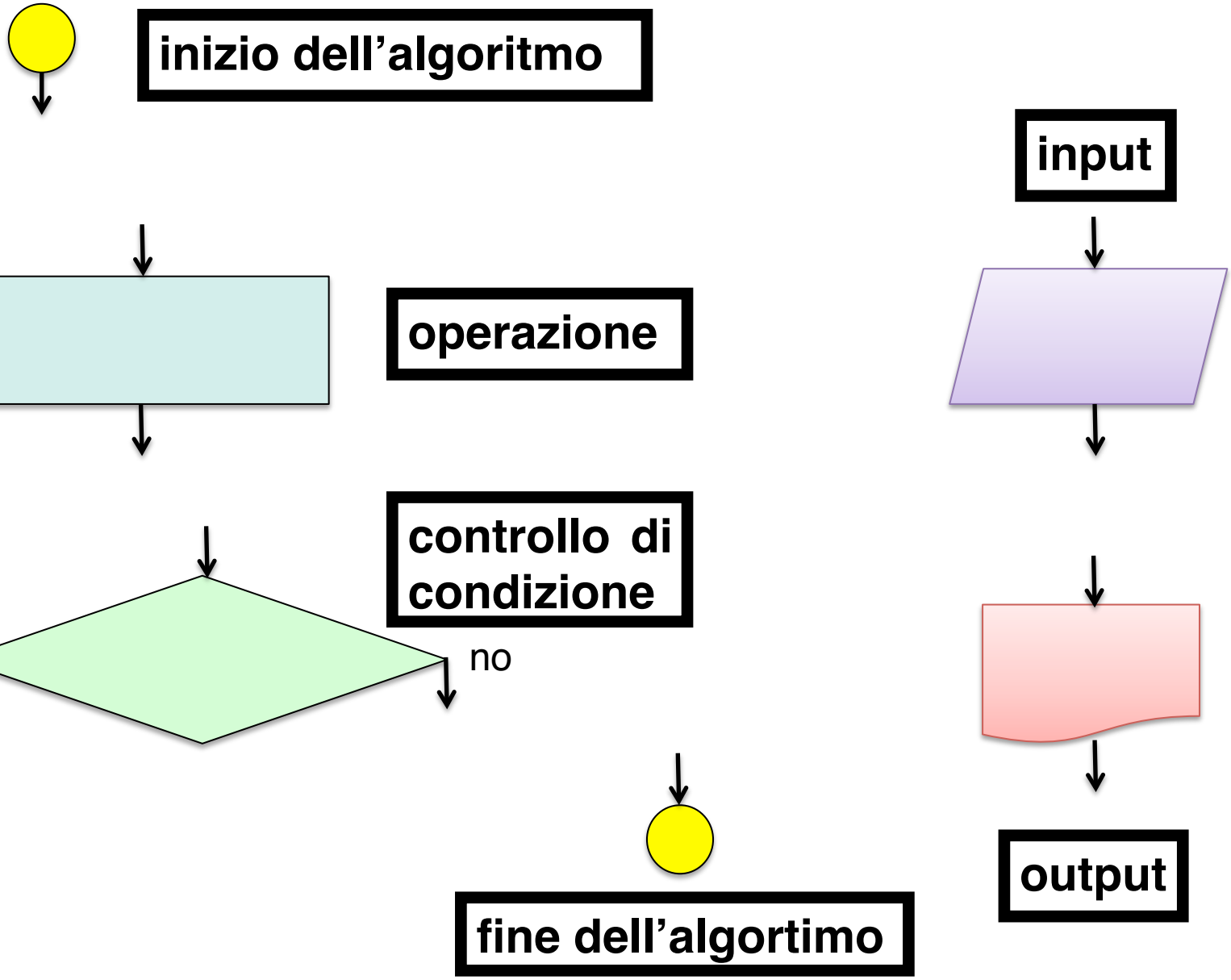
Diagrammi di flusso

Diagramma di flusso

Notazione grafica per esprimere gli algoritmi.

Esiste un elemento grafico (chiamato blocco) per ogni passaggio fondamentale negli algoritmi.

I blocchi sono tra loro collegati da delle freccie, che simboleggiano il flusso di esecuzione, ossia la sequenza di operazioni che vengono svolte una dopo l'altra.


inizio dell'algoritmo
operazione
controllo di condizione
sì
no
fine dell'algortimo
input
output

La freccia entrante in un blocco rappresenta l'istante in cui l'operazione descritta dal blocco sta per essere eseguita.

All'interno di ogni blocco, l'operazione da eseguire è descritta in un linguaggio a scelta, purché compreso dal destinatario del diagramma di flusso.

La freccia uscente da un blocco rappresenta l'istante in cui l'operazione descritta dal blocco è stata eseguita e si passa a quella successiva.

Tutti i blocchi hanno una o più frecce in ingresso e una sola in uscita, tranne il blocco di inizio (zero entranti, una sola uscente), il blocco di fine (una o più entranti, zero uscenti) e il blocco di controllo (una o più entranti, due uscenti: una per ciascun possibile esito del controllo).

I cerchi gialli rappresentano l'inizio dell'algoritmo e la fine dell'algoritmo. Per mantenere il determinismo di un algoritmo ci può essere solo un inizio, mentre ci possono essere più blocchi di fine (a seconda del percorso che è stato seguito).

Un'operazione da eseguire è descritta all'interno di un blocco rettangolare, a meno che non si tratti di un tipo speciale di operazione: input (parallelogramma), output (foglietto), o controllo di condizione (rombo).

Input: informazione oppure entità più concrete arrivano dall'esterno del contesto dell'esecuzione dell'algoritmo per poter essere elaborate.

Output: un risultato (parziale o finale) dell'algoritmo viene inviato verso l'esterno per essere fruito dagli utenti del sistema.

Controllo di condizione: una condizione viene verificata e a seconda del risultato della verifica l'algoritmo segue un percorso o l'altro.

Esercizio 1

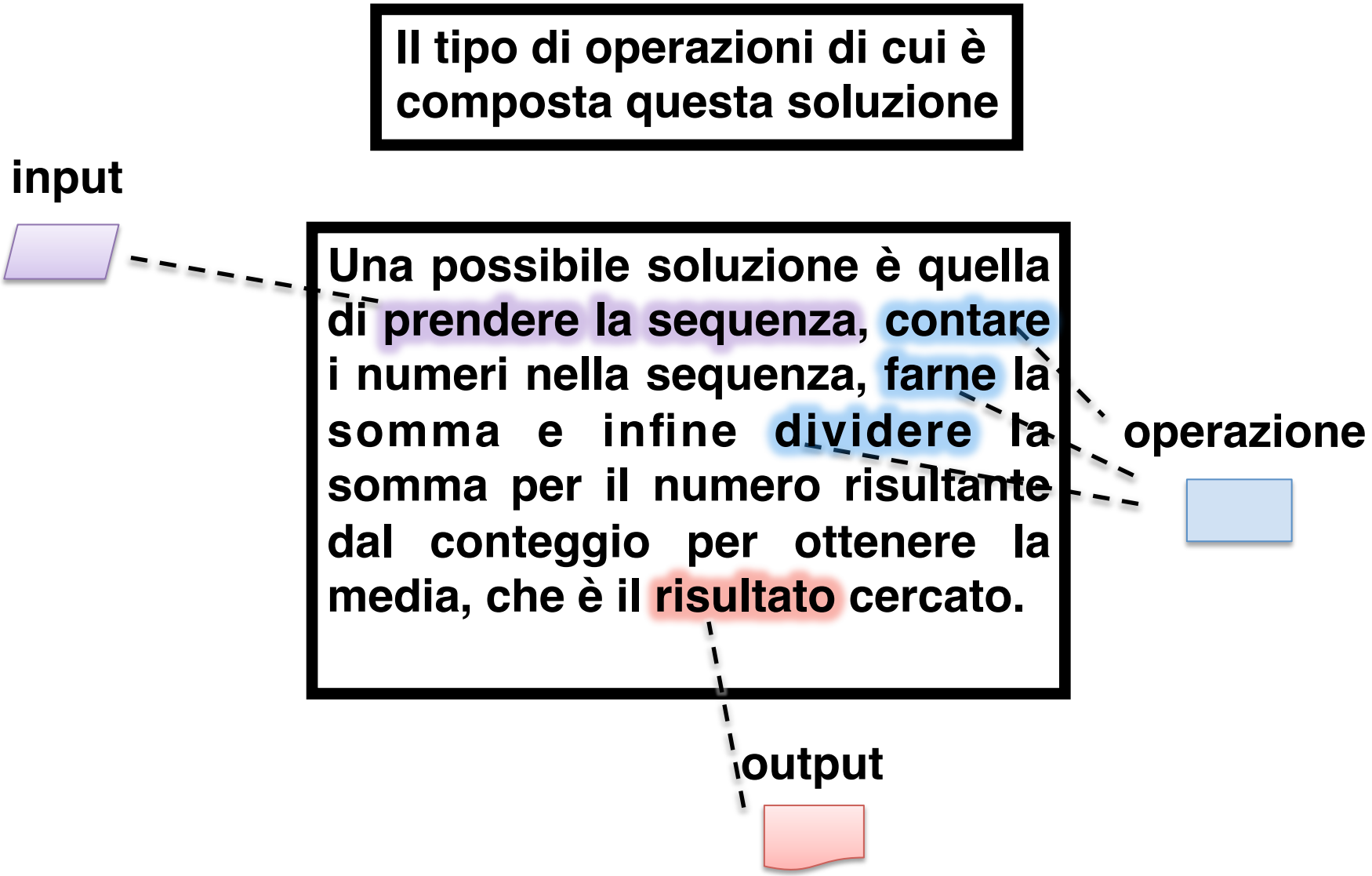
Data una sequenza di numeri, calcolarne la media.

Prima di lanciarci nella soluzione del problema, dobbiamo ricordarci che stiamo elaborando un algoritmo, destinato ad essere trasformato in un programma eseguito da un computer.
L'informazione che viene elaborata nella soluzione, quindi, deve essere organizzata come sequenza di simboli o numeri. Tale organizzazione dei dati deve essere presa in considerazione non solo per l'informazione in input e quella in output, ma anche per i dati intermedi che vengono usati nel corso dell'algoritmo per arrivare alla soluzione.

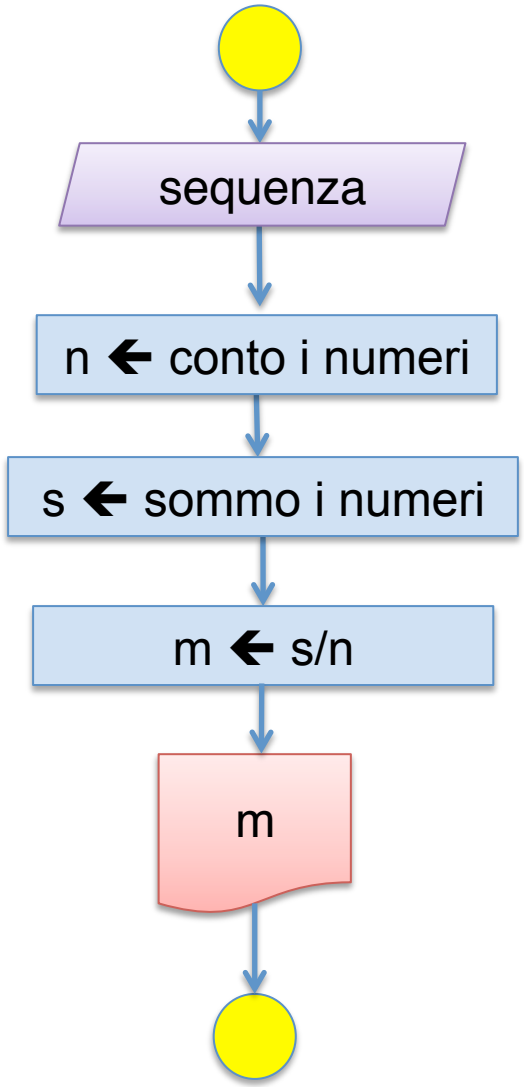
Che siano in input, intermedi, o in output, i dati in un algoritmo possono essere semplici (un singolo numero, un singolo simbolo), oppure strutturati (sequenze di numeri, sequenze di simboli). Un dato strutturato linearmente si chiama array, mentre un dato strutturato bidimensionalmente come una tabella si chiama matrice.

Soluzione

Una possibile soluzione è quella di prendere la sequenza, contare i numeri nella sequenza, farne la somma e infine dividere la somma per il numero risultante dal conteggio per ottenere la media, che è il risultato cercato.


Il tipo di operazioni di cui è composta questa soluzione
input
Una possibile soluzione è quella di prendere la sequenza, contare i numeri nella sequenza, farne la somma e infine dividere la somma per il numero risultante dal conteggio per ottenere la media, che è il risultato cercato.
operazione
output

Algoritmo per calcolare la media dei numeri in una sequenza



s ← sommo i numeri

Un'operazione del tipo mostrato sopra è complessa perché in realtà include più di un passaggio. Il primo passaggio è quello descritto a destra della freccia e indica un'operazione che tipicamente produce un risultato. La freccia rappresenta il secondo passaggio: il risultato ottenuto deve essere conservato per poter essere utilizzato in un momento successivo. La lettera o il nome a sinistra delle freccia rappresenta il "luogo" dove il risultato viene salvato e può essere utilizzato come riferimento per richiamarlo.

In informatica, i "luoghi" dove vengono salvati i risultati delle operazioni si chiamano variabili. Un termine mutuato dalla matematica che vuole indicare che tale luogo può contenere, a seconda delle situazioni, valori differenti, variabili e non costanti.

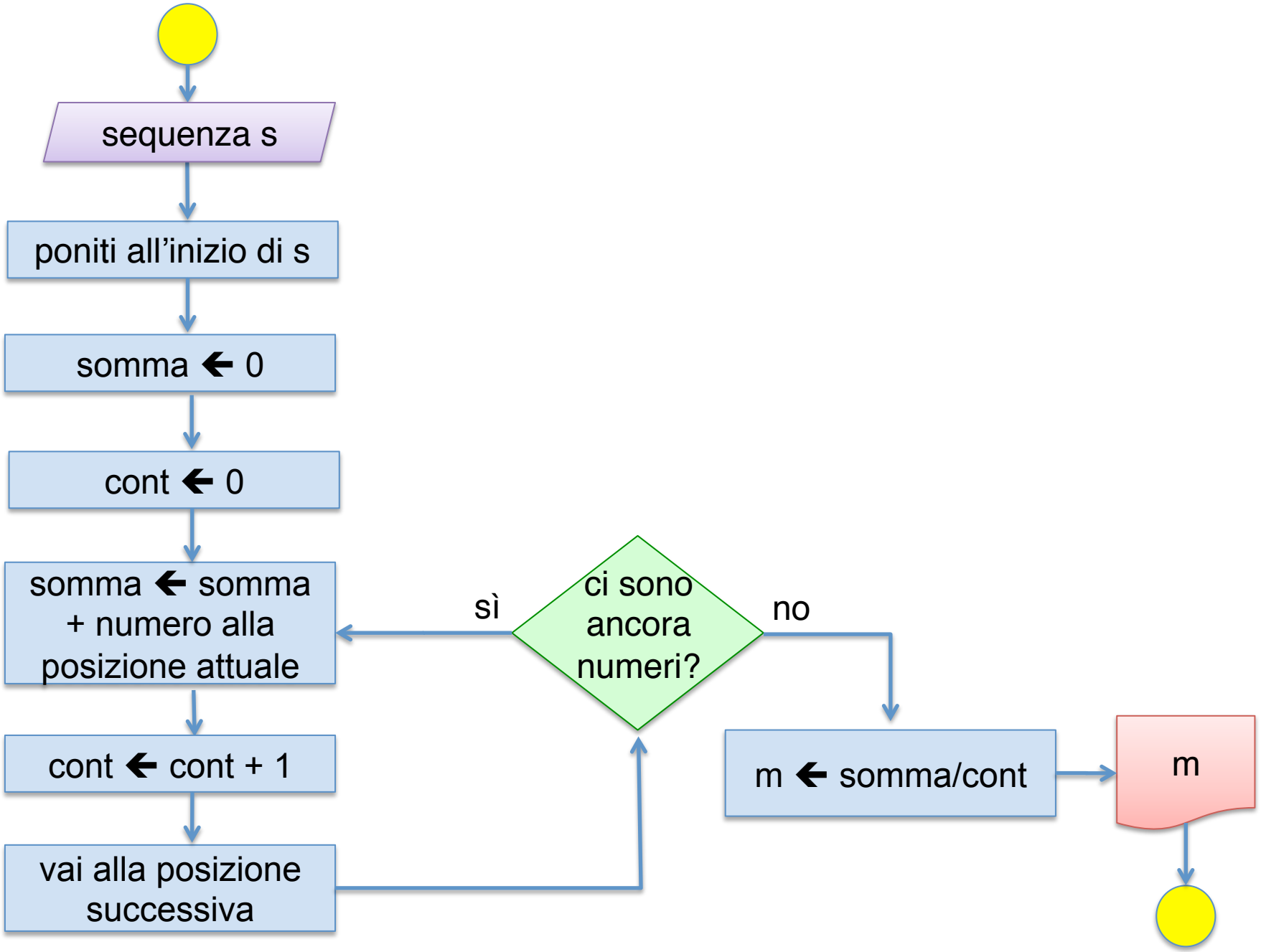
Variabili

Nel caso dell'algoritmo del calcolo della media dei numeri abbiamo usato tre variabili.
n per conservare il numero dei numeri nella sequenza;
s per conservare la somma dei numeri;
m per conservare il valore della media, ottenuta richiamando i valori di s e n e facendo una divisione tra loro.

Come ultima operazione dell'algoritmo, visto che l'obiettivo era quello di calcolare la media, richiamiamo il valore di m e lo mandiamo in output.

Ricordarsi sempre che una soluzione algoritmica a un problema non è l'unica soluzione possibile: si possono eseguire diverse operazioni che portano allo stesso risultato (ad esempio 2 x 3 oppure 3 x 2 danno sempre e comunque 6 come risultato), oppure lo stesso algoritmo può essere descritto a livelli diversi di dettaglio.

Un'alternativa più dettagliata alla soluzione mostrata, ad esempio, può specificare di porsi all'inizio della sequenza dei numeri, controllare che vi sia ancora un numero da conteggiare, incrementare la somma e il conteggio volta per volta e passare alla posizione successiva ripetendo le operazioni precedenti fino a che non ci sono più numeri.


sequenza s
poniti all'inizio di s
somma ← 0
cont ← 0
somma ← somma + numero alla posizione attuale
cont ← cont + 1
vai alla posizione successiva
ci sono ancora numeri?
sì
no
m ← somma/cont
m

Da notare, in quest'ultima soluzione, che il flusso che porta dall'inizio alla fine dell'algoritmo non è lineare: a un certo punto c'è un controllo di condizione e, a seconda del risultato, si torna indietro nel percorso per ripetere delle operazioni.

La ripetizione di operazioni in un algoritmo è tipicamente rappresentata nel diagramma di flusso corrispondente da un percorso circolare, motivo per cui le ripetizioni negli algoritmi si chiamano anche cicli, oppure loop (in inglese).

Naturalmente, è fondamentale che la ripetizione abbia un termine, altrimenti la fine dell'algoritmo non si raggiungerebbe mai: sarà cura della persona che concepisce la soluzione di accertarsi che la condizione di uscita dal ciclo si realizzi a un certo punto dell'esecuzione.

Struttura dei dati

Tutti i dati (quelli in input, quelli usati all'interno di un algoritmo, quelli in output) sono organizzati sotto forma di strutture

Tali strutture possono essere semplici per dati singoli (come le variabili) oppure più complesse per dati multipli (come gli array o le matrici

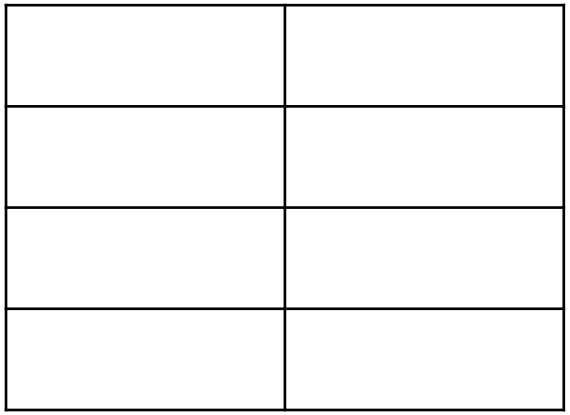
È compito di chi concepisce l'algoritmo trovare le strutture più adatte per ospitare i dati elaborati dall'algoritmo stesso

Esercizio 2

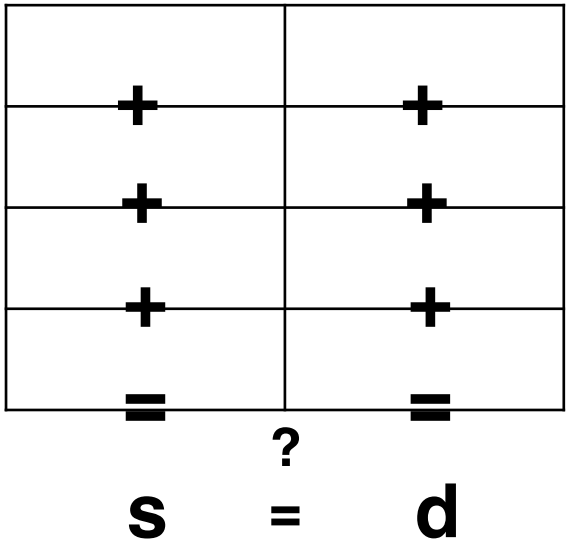
Data la lista dei pesi dei passeggeri su un aereo a 8 posti, scrivere l'algoritmo che dica se il peso a sinistra e quello a destra del velivolo sono uguali o meno.

Struttura dei dati

In questo caso, possiamo usare un modello astratto della disposizione dei posti nel velivolo come struttura dei dati in input: se ci sono 8 posti, possiamo immaginare di organizzare i pesi in una matrice con 4 righe e 2 colonne



L’algoritmo dovrà:
- sommare i valori presenti sulla prima colonna per avere il peso totale a sinistra
- sommare i valori presenti sulla seconda colonna per avere il peso totale a destra
- confrontare le due somme e inviare l’esito del confronto in output



Abbiamo già detto che se esiste una soluzione algoritmica a un problema, ve ne sono numerose altre che risolvono il problema.

Per gli algoritmi, però, non vale il discorso di "uno vale l'altro": ci sono delle caratteritische che rendono un algoritmo migliore di un altro.

Di base, tutti gli algoritmi per risolvere un problema devono essere corretti : un algoritmo si dice corretto quando risolve il problema per il quale è stato concepito. Sembra una caratteristica ovvia ma molto spesso scriviamo algoritmi non corretti a causa di distrazioni, errori, etc.

Altra caratteristica desiderabile anche se non fondamentale come la correttezza è l'efficienza : un algoritmo è più efficiente di un altro se la sua esecuzione consuma meno risorse (in termini di tempo, energia, etc.)

Ora ci concentriamo su una terza caratteristica: la scalabilità.
Un algoritmo si dice scalabile quando si adatta facilmente a input diversi. Vediamo in seguito che cosa vuol dire.

Scalabillità di un algoritmo

Vediamo subito un esempio di algoritmo non scalabile.
Sia la tabella sottostante l'insieme di dati in input per il problema del bilanciamento dei pesi.

56	87
79	60
23	75
88	49

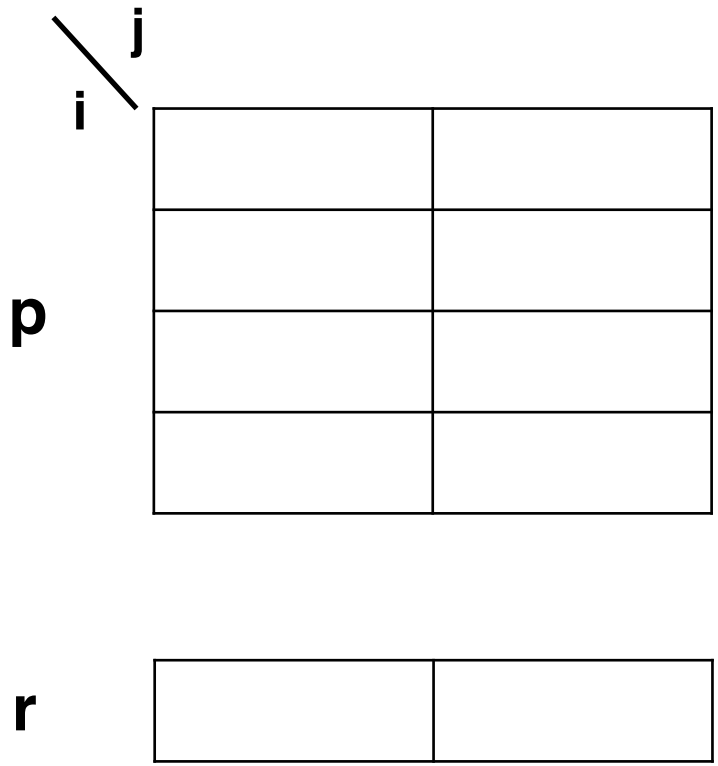
Un possibile algoritmo che risolve il problema è il seguente:
56 + 79 + 23 + 88 = 246
87 + 60 + 75 + 49 = 271
246 ≠ 271, quindi output: NO

Questo algoritmo non è scalabile perché le sue operazioni sono basate esplicitamente sui valori che arrivano in input.
È vero che ci fornisce la soluzione cercata, ma funziona solo con l'input 56, 79, 23… Se arrivassero in input dati diversi, l'algoritmo non funzionerebbe.

Un modo per incrementare la scalabilità è quello non di basarsi sui valori numerici dei pesi, ma di fare riferimento alle posizioni dei questi valori nella tabella. In questo modo, qualunque siano i numeri in input, le operazioni dell'algoritmo funzionano correttamente.

Se si lavora con variabili semplici, si usa il loro nome per fare riferimento al valore in esse contenuto.
Con gli array, introduciamo la notazione p[k] per fare riferimento al valore contenuto nella posizione k dell'array p.
Con le matrici, scriviamo q[i][j] per indicare il valore presente all'incrocio tra la riga i e la colonna j della matrice q.

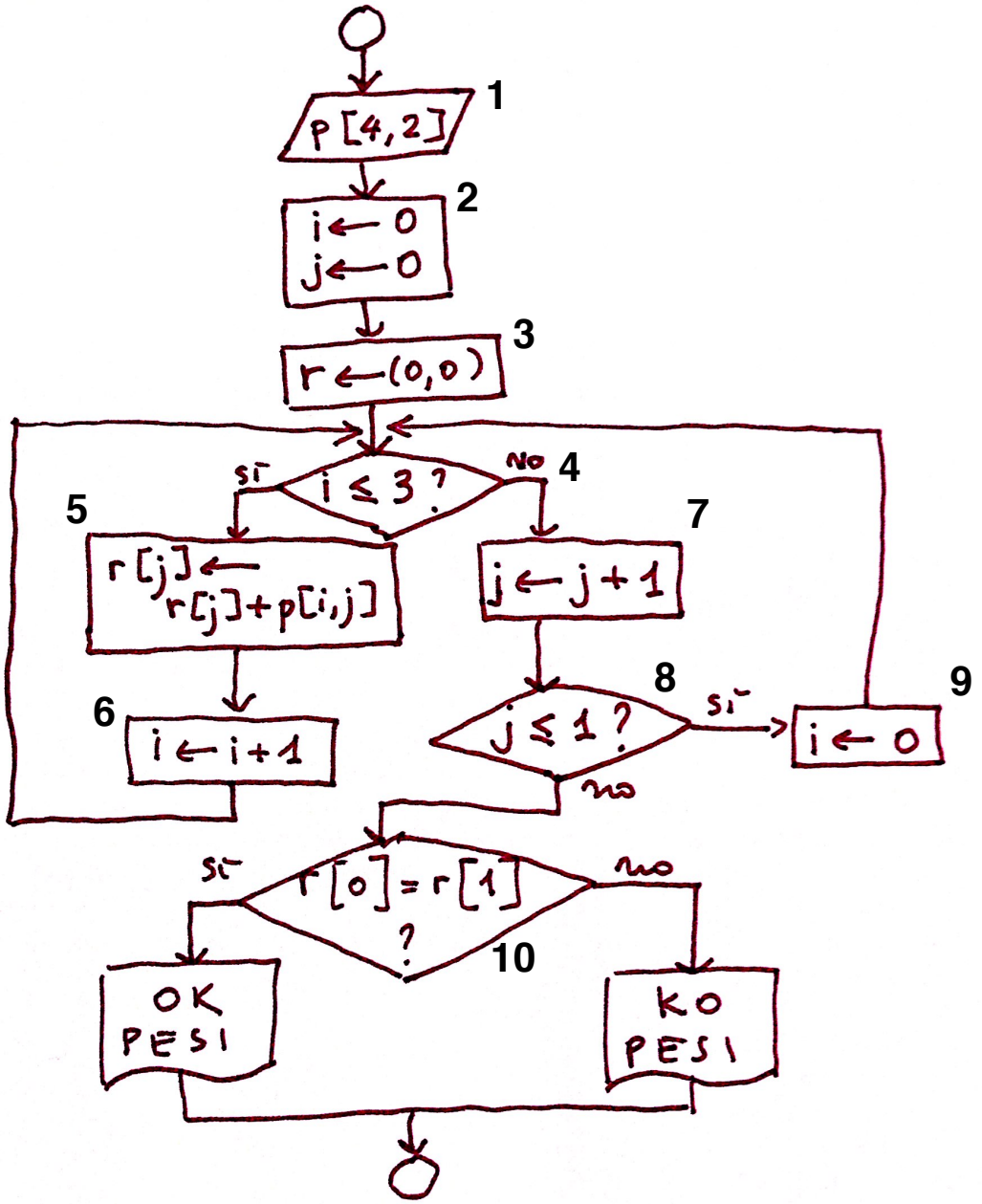
Strutture dati per l'algoritmo



Utilizziamo una matrice p 4x2 per i pesi in input. Useremo una variabile i per indicare la riga della matrice con cui stiamo lavorando e una variabile j per indicare la colonna. In questo problema, i è compreso tra 0 e 3 (tipicamente in informatica si inizia a contare da 0), j è compreso tra 0 e 1.

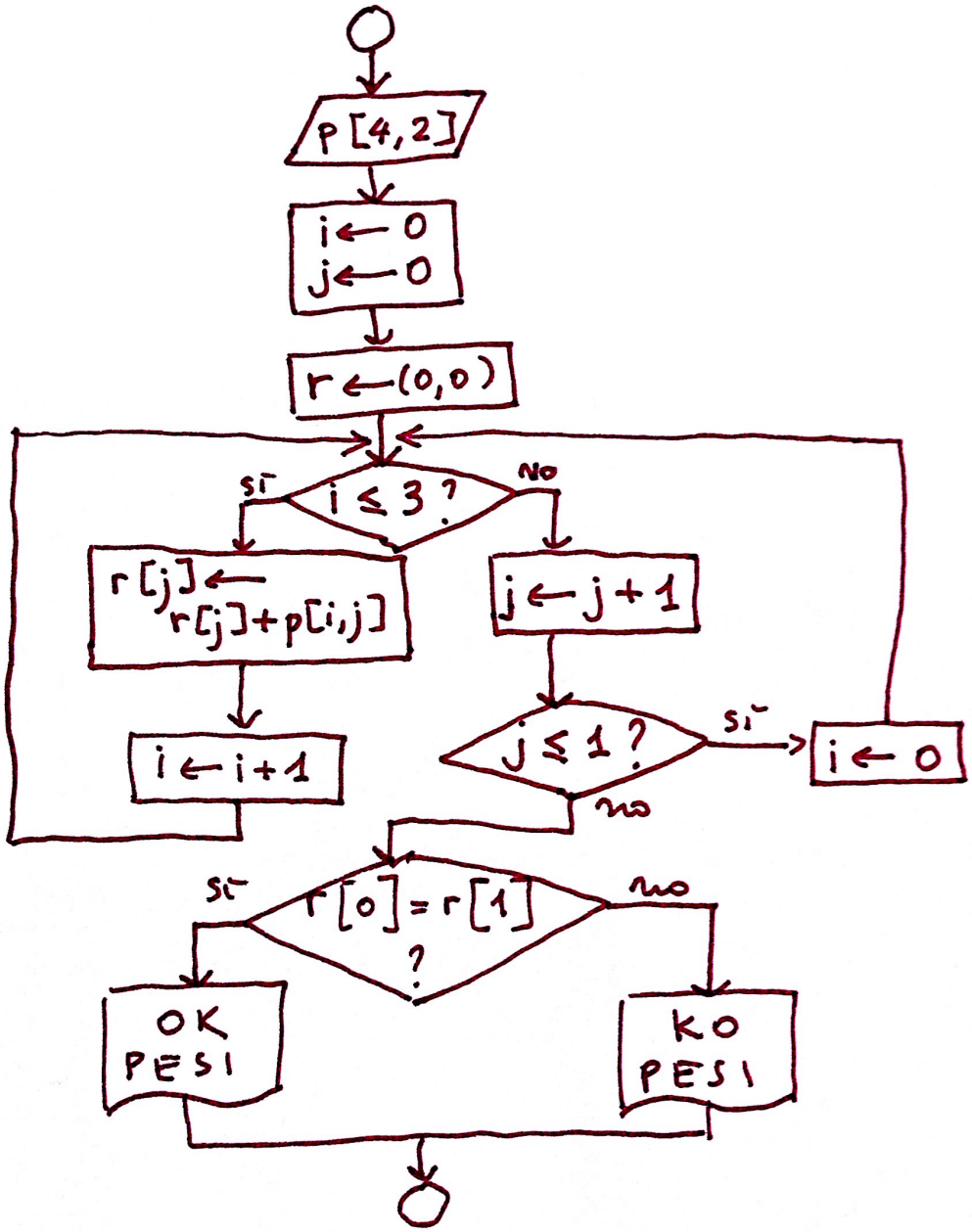
La somma dei pesi di ciascuna colonna di p verrà salvata in un'apposita casella di un array r.

Diagramma di flusso dell'algoritmo



Note sull'algoritmo

1. Quando arriva in input un dato strutturato, possiamo specificarne le dimensioni tra parentesi quadre.
2. L'assegnamento di un valore iniziale a delle variabili si chiama inizializzazione. In questo caso sia i sia j vengono inizializzate a 0.
3. r è un array di dimensione due, quindi la sua inizializzazione richiede due valori iniziali. Visto che conterrà le somme dei pesi delle due colonne, i suoi valori iniziali sono entrambi 0, e ad essi andranno a sommarsi i vari pesi.
4. Si tratta di un controllo per vedere se siamo già arrivati in fondo alla tabella dei pesi p. Visto che l'ultima riga è la n.3, e usiamo i come indice della riga, se i non ha superato 3, vuol dire che non abbiamo ancora finito di scorrere le righe di p.
5. Se stiamo lavorando con la colonna j di p, dobbiamo andare a sommare i pesi di questa colonna nella posizione j di r. All'inizio j vale 0, quindi andiamo a sommare i pesi della colonna 0 nella posizione 0 di r.
6. Sommato il peso alla riga i, dobbiamo passare alla riga successiva, quindi aumentiamo i di 1 e torniamo a controllare che non siano finite le righe.
7. Se le righe sono finite (cioè r > 3) allora dobbiamo passare alla colonna successiva, e per fare ciò incrementiamo j (indice della colonna) di 1.
8. Se j non ha superato 1, vuol dire che c'è ancora una colonna su cui lavorare…
9. …e quindi resettiamo i a 0, in modo da lavorare sulla colonna dalla prima riga.
10. Se invece j ha superato 1, vuol dire che abbiamo finito con le colonne, quindi abbiamo fatto tutte le somme che dovevamo fare e possiamo passare al confronto delle somme, ossia dei valori in p: se sono uguali i pesi sono bilanciati, altrimenti non lo sono.



Questo algoritmo gode della caratteristica di essere scalabile.
Infatti, dovesse arrivare una tabella dei pesi con valori diversi funzionerebbe lo stesso.
Se anche dovessimo lavorare con una tabella di dimensioni diverse, le modifiche da fare sarebbero minime.
Sapete dire quali?