



Informatica

Capitolo 1 – Introduzione

Prof. Ivan Gentile

Problema: Media voti delle materie

- Voglio Calcolare la media dei voti di uno studenti alla fine dell'anno scolastico.
- INPUT:
 - N: numero di materie
 - Voti
- Output
 - M: media dei voti
- Problema: Quanti sono i voti? Quante variabili devo dichiarare?
 - Devo dichiarare n variabili, ma non so come fare perché n è un valore che devo acquisire
 - Inoltre se n fosse molto grande dovrei usare tante variabili che non sono indispensabili per calcolare l'output

$$M = (V1 + V2 + \dots + VN) / N$$

Soluzione

- Posso usare una sola variabile voto V
- Il nuovo voto sostituirà il vecchio
- Mi costruisco mano mano la somma S
- Algoritmo:
 - Parto da $S = 0$
 - Leggo il voto V
 - $S = S + V$
 - Devo ripetere le due operazioni precedenti per N volte
- Mi serve un meccanismo per ripetere delle operazioni: esiste e si chiama **Iterazione** o **Ciclo**



Fare il flowchart in flowgorithm

teorema di Böhm-Jacopini

- Enunciato nel 1966 dagli informatici Corrado Böhm e Giuseppe Jacopini,
- Afferma che ***"qualunque algoritmo può essere implementato in fase di programmazione (in diagramma di flusso, pseudocodice o codice sorgente) utilizzando tre sole strutture dette strutture di controllo: la sequenza, la selezione e l'iterazione"***

Divisione con sole sottrazioni

- La divisione intera tra due numeri restituisce due valori
 - Quoziente, Resto
 - Esempio: $7 / 3 \Rightarrow Q = 2, R = 1$
- Vogliamo realizzare la divisione con sole sottrazioni

Riflettiamo sul significato di divisione (intera)

- Dividere 7 per 3 ($7 / 3$) significa che abbiamo 7 oggetti (es. caramelle) e vogliamo distribuirle a 3 persone
- Vogliamo però che ognuna delle 3 persone abbia lo stesso numero di caramelle
- Il numero di distribuzioni che riesco a fare è il quoziente, quelle che mi avanzano è il resto
- Es:
 - Inizio a distribuire una caramella ciascuno, mi avanzano 4 ($7 - 3$) caramelle
 - Posso ancora distribuire una caramella ciascuno, mi avanzano 1 ($4 - 3$) caramelle
 - Non posso più distribuire caramelle, mi è avanzata 1 caramella quindi il resto è 1
 - Ho fatto due distribuzioni: il quoziente è 2

Algoritmo

- INPUT: Dividendo (DVD) e Divisore (DVS)
- OUTPUT: Quoziente (Q), Resto (R)
- Algoritmo:
 - Sottrarre DVD a DVS finché $DVD \geq DVS$
 - Il numero delle sottrazioni fatte è Q
 - Il valore finale di DVD è R



Fare il flowchart in flowgorithm

Massimo Comune Divisore

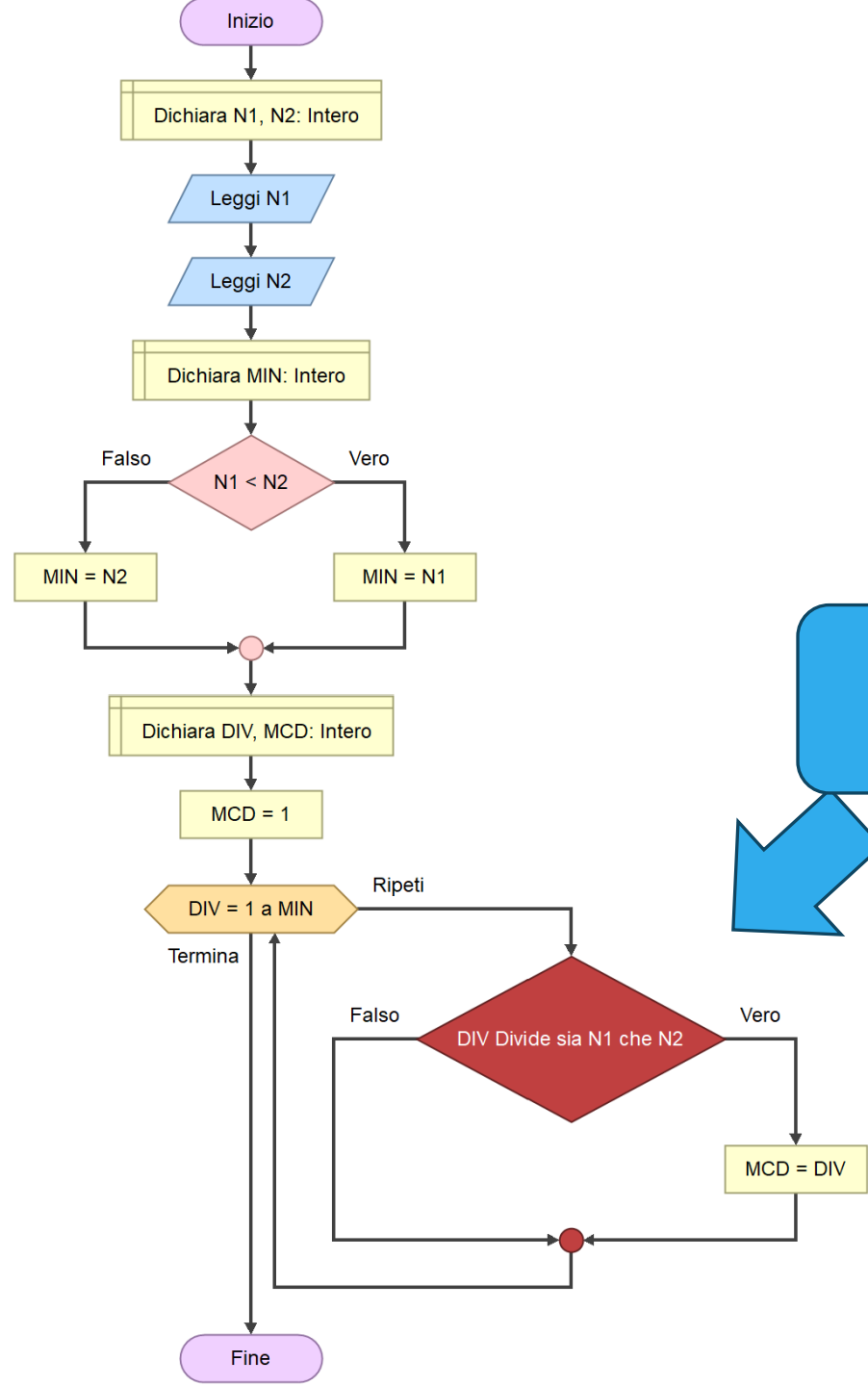
- Divisore di numero N (intero) è un numero DVS (intero) tale che la divisione tra N e DVS dà resto zero
- Ogni numero ha due divisori banali 1 e se stesso
- Massimo Comune Divisore (MCD) tra due numero X e Y :
 - Si calcolano tutti i divisori di X e tutti i divisori di Y
 - Si prende il valore più grande che si trova sia tra i divisori di X che di Y

MCD: Esempio

- $N_1 = 12, N_2 = 8$
- Tutti i divisori di 12: 1, 2, 4, 6, 12
- Tutti i divisori di 8: 1, 2, 4, 8
- Il divisore più grande comune ai due insiemi è 4 (MCD)

Algoritmo

- Non ho possibilità di memorizzare tutti i divisori dei due numeri
 - Non so quanti ne sono e non saprei dove memorizzarli
- Devo ristrutturare il problema:
 - Parto da 1 (che sicuramente è un divisore comune) e incremento di 1
 - Ogni numero che divide entrambi è il numero potenziale massimo comune divisore
 - Ci fermiamo quando siamo arrivati al minimo tra N_1 e N_2 (perché il MCD non può mai superare il minimo tra essi)



Da Sviluppare



Fare il flowchart in flowgorithm

Multipli e Minimo Comune Multiplo (MCM)

- Dato un numero intero X un suo **multiplo** è un qualunque numero che si ottiene dalla moltiplicazione di X per un intero N (≥ 1)
- I multipli di 12 e 8
 - Multipli di 12: 12, **24**, 36, 48, 60, 72, ...
 - Multipli di 8: 8, 16, **24**, 32, 40, 48, ...
- I multipli di un numero sono **infiniti**
- Tra tutti gli infiniti multipli di due numeri (o più numeri) il **MCM** è il più piccolo tra quelli comuni
 - $\text{MCM}(12, 8) = 24$

Algoritmo

- Dovremmo memorizzare tanti multipli (addirittura infiniti) e poi trovare il minimo comune ai due insiemi
- Non abbiamo gli strumenti per memorizzare un numero imprecisato di variabili
- Bisogna ripensare l'algoritmo

Algoritmo

- Calcolo di volta in volta i multipli di N_1 e N_2
 - $M_1 = N_1 * MC_1$
 - $M_2 = N_2 * MC_2$
 - dove MC_1 e MC_2 sono i moltiplicatori, interi che partono da 1 e vengono incrementati di 1
- Se $M_1 < M_2$ incremento MC_1 , altrimenti MC_2
- Il processo termina quando $M_1 = M_2$, questo è il MCM

Esempio

- $N1 = 4, N2 = 6$
- $MC1 = MC2 = 1$
- $M1 = N1 * MC1 = 4 * 1 = 4$
- $M2 = N2 * MC2 = 6 * 1 = 6$
- $M1$ è minore di $M2$ incremento $MC1$, dunque : $MC1 = MC1 + 1 = 1 + 1 = 2$
- $M1 = N1 * MC1 = 4 * 2 = 8$
- $M1$ è maggiore di $M2$ incremento $MC2$, dunque: $MC2 = MC2 + 1 = 1 + 1 = 2$
- $M2 = N2 * MC2 = 6 * 2 = 12$
- $M1$ è minore di $M2$ incremento $MC1$, dunque : $MC1 = MC1 + 1 = 2 + 1 = 3$
- $M1 = N1 * MC1 = 4 * 3 = 12$
- $M1$ ed $M2$ sono uguali, tale valore (12) è il MCM



Fare il flowchart in flowgorithm

Altra soluzione

- Il MCM è sempre \geq del massimo tra i due numeri
- Diciamo MAX il massimo tra N_1 e N_2 e MIN il minimo
- Se il resto della divisione tra MAX e MIN è zero allora $MCM = MIN$
- Se non è così calcolo tutti i multipli di MAX fino a trovare quello il cui resto della divisione con MIN dà 0
- Nota: potrei partire da MIN ma non conviene perché comunque devo a forza di moltiplicazioni superare il MAX e se la differenza tra MAX e MIN è molto alta ci possono volere molte istruzioni e molto codice.



Fare il flowchart in flowgorithm

Conversione da binario a decimale (bin2dec)

- Vogliamo convertire un numero da decimale (base 10) a binario (base 2)
 - Esempio: 1100 -> 12
- Il sistema binario (come il decimale) è **posizionale pesato**
 - $1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = 8 + 4 + 0 + 0 = 12$

Analisi

- I nostri tipi primitivi sono intero, reale, booleano, carattere
 - (Flogorithm ha il tipo stringa ma nei linguaggi di programmazione non è tipicamente un tipo primitivo)
- I tipi numerici sono in base 10 e non in base due,
 - Se scrivo 1100 il computer pensa che sia millecento (e non "uno-uno-zero-zero" che corrisponde a 12)

Analisi

- Devo allora fornire il numero binario in input una **cifra alla volta**
 - Tanto 1 e 0 in base due corrispondono 1 e 0 in base dieci
- Supponiamo che non vogliamo che il numero di bit sia stabilito a priori. In tal caso:
 - Bisogna chiedere all'utente ogni volta se vuole inserire una nuova cifra binaria o meno.
 - Il ciclo sarà un ciclo **indefinito**
 - Useremo allora una **variabile booleana**
- **Altra domanda ci facciamo dare le cifre dalla MSB alla LSB (da sinistra a destra) o dalla LSB alla MSB (da destra a sinistra)?**
 - **Convieni dalla LSB alla MSB perché l'esponente della potenza di 2 sarà da 0 a crescere**
 - mentre se partiamo dal MSB non sappiamo che esponente dargli perché non sappiamo l'utente quante cifre vorrà inserire



Fare il flowchart in flowgorithm

Se la potenza (2^i) non esiste

- La potenza è una moltiplicazione ripetuta
 - 2^i significa moltiplicare 2 per se stesso $i-1$ volte (in modo che nel prodotto 2 compaia i volte)
 - Dovremmo usare un ciclo definito
- Nel nostro algoritmo avremo quindi un ciclo nel ciclo
- Si può fare, ma se ristrutturiamo il nostro problema possiamo evitare il ciclo innestato
 - basta scrivere una formula iterativa per la potenza di 2
 - $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, il valore della potenza è il doppio di quello precedente. Se parto da $m = 1$, la formula sarà $m = m * 2$



Fare il flowchart in flowgorithm

Analisi

- E' scomodo chiede ogni volta se inserire una nuova cifra o meno
- Possiamo raggruppare in una sola richiesta la cifra e se vogliamo continuare
 - Visto che la variabile bit è un tipo intero può avere più valori
 - 1 rappresenta il bit 1
 - 0 rappresenta il bit 0
 - 2 se vogliamo terminare



Fare il flowchart in flowgorithm

Nuove sfide

- **Se volessimo dare le cifre partendo dal MSB come potremmo fare?**
- **Potremmo dare tutte le cifre insieme?**
 - cioè anche se dichiariamo una variabile intera in base 10 (perché non possiamo dichiarare variabili in altre basi con le conoscenze in cui abbiamo) diamo il numero tutto insieme (esempio 1011) questo numero lo consideriamo binario e non decimale

Numero Primo

- Dato un numero intero positivo stabilire se è primo

Analisi

- Un numero primo è un numero che non si può scomporre nel prodotto di due fattori che non siano banali (i fattori banali sono 1 e il numero stesso)
 - 12 NON è primo perché si scomporre in due fattori come 2 e 6 oppure 4 e 3 ($2 \cdot 6 = 12$, $4 \cdot 3 = 12$)
 - 11 è primo perché non si può scomporre in fattori, tranne quelli banali ($11 = 1 \cdot 11$)
- Se a partire da 2 fino a $N-1$ (compreso) troviamo un numero che divide N possiamo dire che N NON è primo



Fare il flowchart in flowgorithm

Osservazione

- Quando trova il primo divisore (e quindi siamo certi che il numero non è primo) continua ad andare avanti fino ad $N-1$
 - Se il numero è 3000 000 000 il primo divisore 2 ma continua a fare miliardi di iterazioni prima di uscire
- Per evitare ciò basta forzare l'uscita
 - mettere un valore ad i maggiore dell'estremo superiore del ciclo, esempio $i = N$
 - basta anche mettere $i=N-1$ perché ricordiamo che il for prima di verificare la condizione fa l'incremento di n



Fare il flowchart in flowgorithm