



Università degli Studi di Verona

Dipartimento di Biotecnologie

Laurea in Biotecnologie

Corso di Informatica 2014/2015

Algoritmi e Diagrammi di flusso

Gennaio 2015 - Sergio Marin Vargas

Problemi e soluzioni

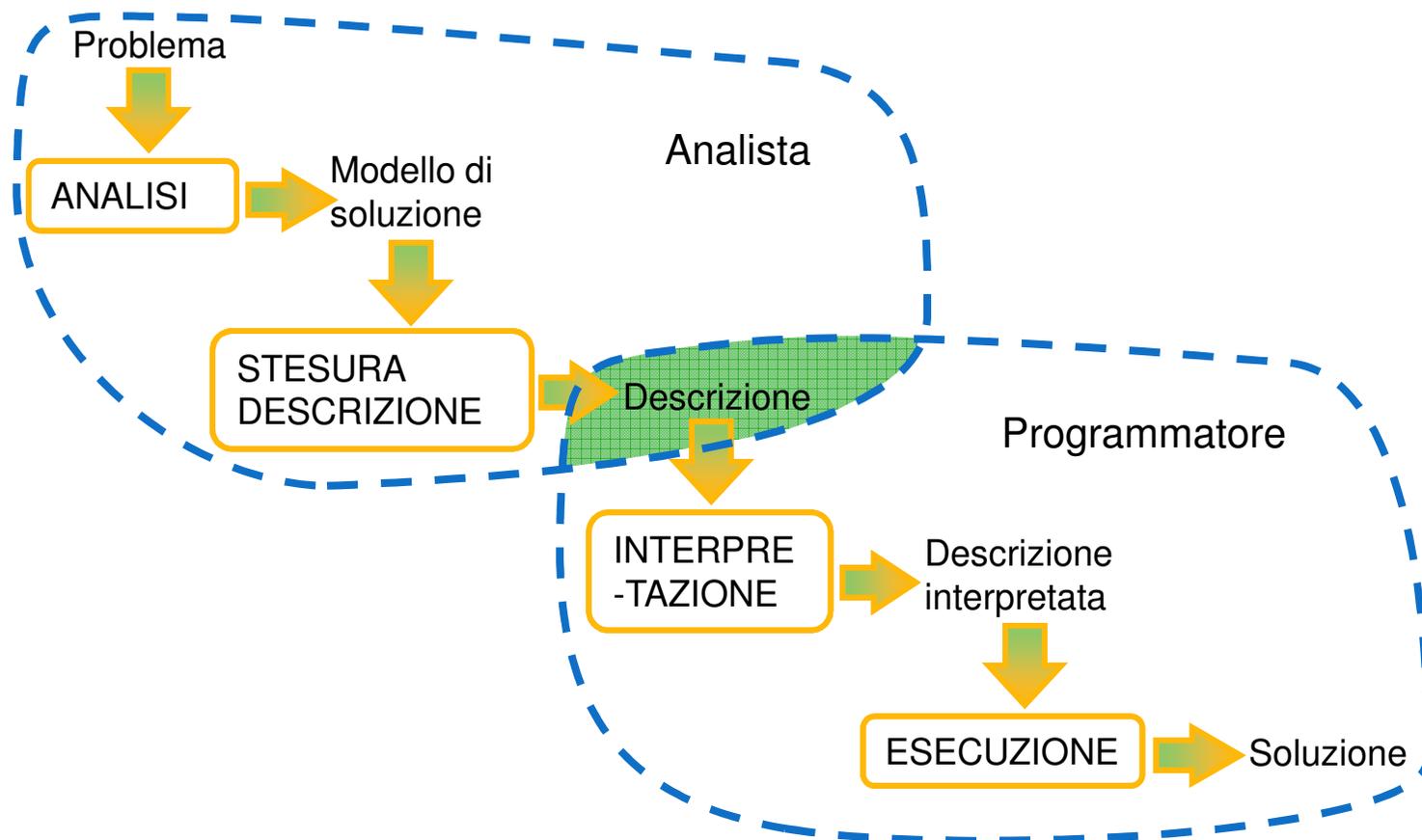
Capacità di descrivere il procedimento di soluzione?

- Descrivere la soluzione come algoritmo
cioè la capacità di **sapere cosa deve fare** l'esecutore
- Descrivere la soluzione come programma
cioè la capacità di **dire all'esecutore esattamente** cosa fare
- Linguaggio interpretabile dall'esecutore
cioè i linguaggi di programmazione disponibile nell'esecutore
- Azioni che l'esecutore è in grado di eseguire
cioè la conoscenza delle istruzioni del linguaggio

IL CALCOLATORE E' UN ESECUTORE

Analista vs Programmatore

Conoscenza di come si risolve un problema
vs
Capacità di risolvere un problema



Definire il problema

- Eliminare le ambiguità nella formulazione del problema
- Individuare il risultato che si vuole ottenere, gli obiettivi da raggiungere
- Evidenziare
 - le regole da rispettare
 - i vincoli interni ed esterni
 - i dati espliciti ed impliciti
- Eliminare i dettagli inutili ed ambigui

Problemi elementari

- La descrizione della soluzione di un problema dipende dall'esecutore, questa deve tener conto dalla sua disponibilità di istruzioni di base
 - si scompone il problema originario in **sottoproblemi**;
 - si scompongono i sottoproblemi in **sotto-sottoproblemi**;
 - si prosegue nella scomposizione fino a giungere a **problemi elementari** (o **primitivi**), cioè problemi che possono essere risolti direttamente dall'esecutore
 - Ogni **problema elementare** è associato a un'**azione elementare** che può essere direttamente compiuta dall'esecutore

Proprietà di un'azione elementare

- **Finitezza**
 - l'azione deve concludersi in un tempo finito
- **Osservabilità**
 - l'azione deve avere un effetto osservabile, cioè deve produrre qualcosa
- **Riproducibilità**
 - a partire dallo stesso stato iniziale, la stessa azione deve produrre sempre lo stesso risultato

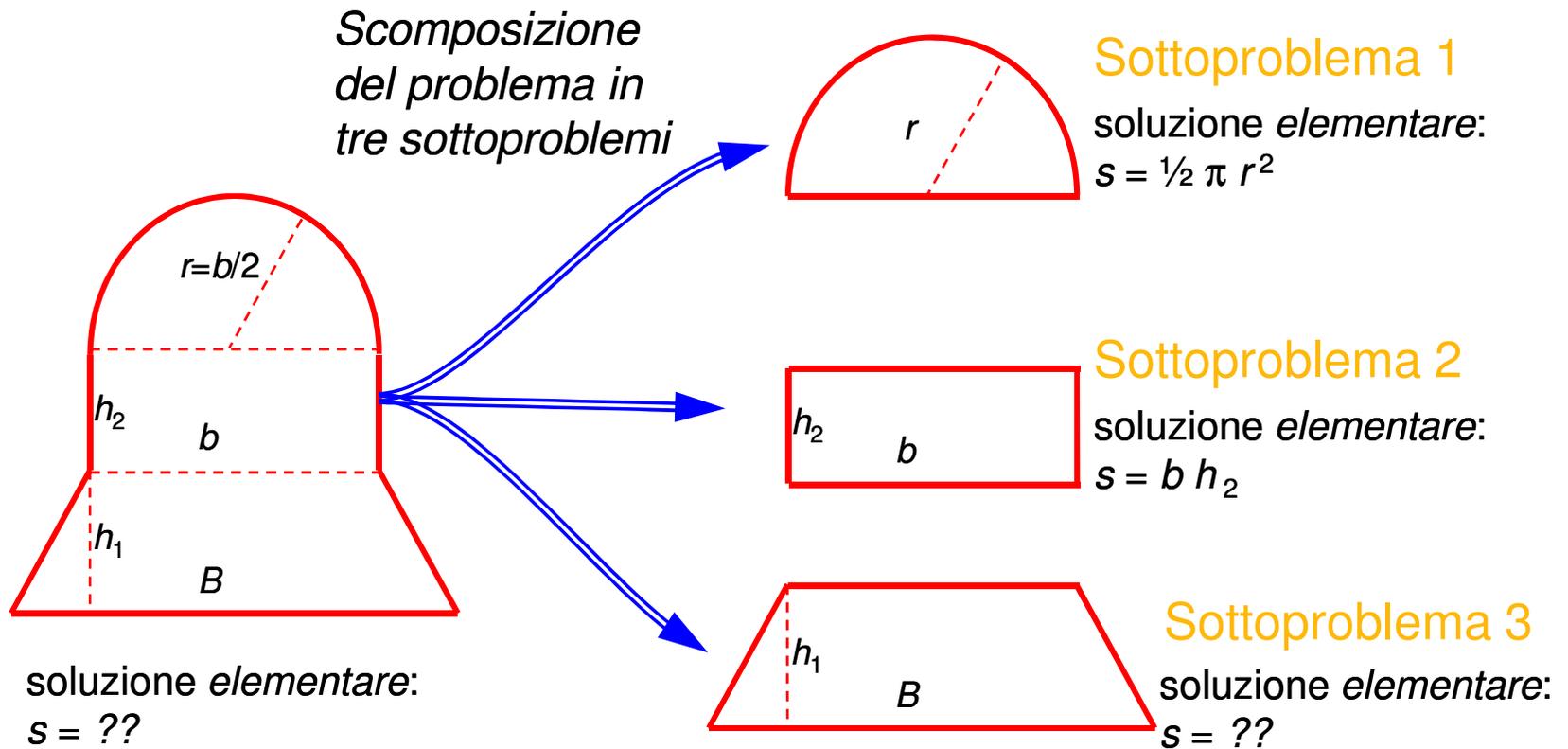
Esempio di azioni elementari

- Operazioni aritmetiche e assegnamenti di valori a singole variabili
 - Es. $C \leftarrow (A + B)$
- Condizioni sul valore di singole variabili
 - se $(A > B)$ allora ... altrimenti ...
- Lettura e scrittura di variabili
 - “Leggi A” oppure “Stampa B”

Esempio di risoluzione (I)

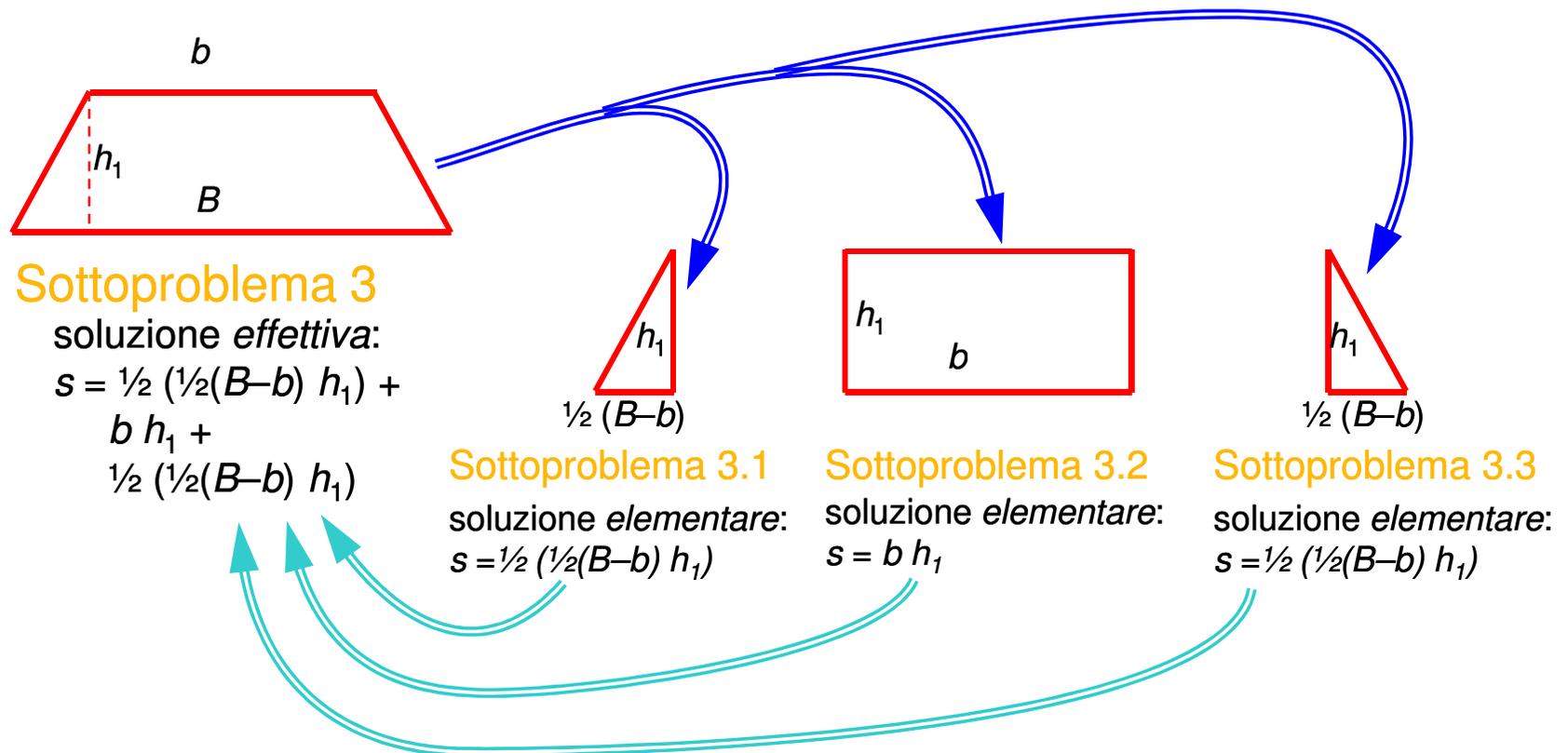
Problema

Calcolare l'area "s" di una campana come quella indicata nella figura



Esempio di risoluzione (2)

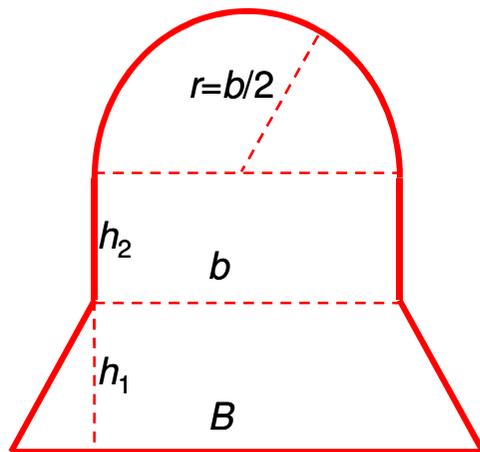
Scomposizione del sottoproblema 3 in tre ulteriori **sotto-sottoproblemi**



Composizione delle soluzioni dei tre sottoproblemi 3.1, 3.2 e 3.3 per risolvere il sottoproblema 3

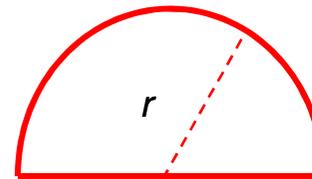
Esempio di risoluzione (3)

Problema



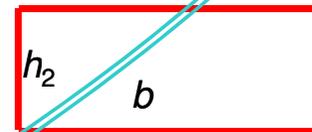
soluzione effettiva:

$$s = \frac{1}{2} \pi r^2 + b h_2 + \frac{1}{2} (\frac{1}{2}(B-b) h_1) + b h_1 + \frac{1}{2} (\frac{1}{2}(B-b) h_1)$$



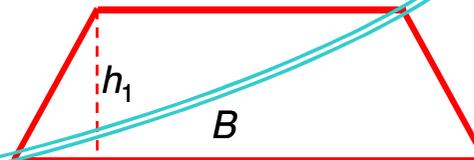
Sottoproblema 1

soluzione elementare:
 $s = \frac{1}{2} \pi r^2$



Sottoproblema 2

soluzione elementare:
 $s = b h_2$



Sottoproblema 3

soluzione effettiva:
 $s = \frac{1}{2} (\frac{1}{2}(B-b) h_1) + b h_1 + \frac{1}{2} (\frac{1}{2}(B-b) h_1)$

Composizione delle soluzioni dei tre sottoproblemi 1, 2 e 3 per risolvere il problema originario



Algoritmo (definizione informale)

- Sequenza **finita** di istruzioni,
- Che possono essere tradotte in un programma **comprensibile** da un esecutore,
- che **describe** come realizzare un compito cioè come risolvere un problema.

Proprietà degli algoritmi

- **Correttezza**
 - L'algoritmo perviene alla soluzione del compito cui è preposto, senza difettare di alcun passo fondamentale
- **Efficienza**
 - L'algoritmo perviene alla soluzione del problema cercando di utilizzare la minima quantità di risorse fisiche
 - tempo di esecuzione,
 - memoria,
 - ...

Riassumendo

- *Algoritmo* = descrizione di come si risolve un problema
- *Programma* = algoritmo scritto in modo che possa essere interpretato e poi eseguito da un calcolatore (linguaggio di programmazione)
- *Linguaggio macchina* = linguaggio effettivamente “compreso” da un calcolatore, caratterizzato da istruzioni primitive semplici, ma che sono difficile e noiose da utilizzare
- Principi base di programmazione:
 - **produrre algoritmi** (cioè capire la sequenza di passi che portano alla soluzione di un problema)
 - **codificarli in programmi** (cioè renderli comprensibili al calcolatore)



Rappresentazione degli algoritmi

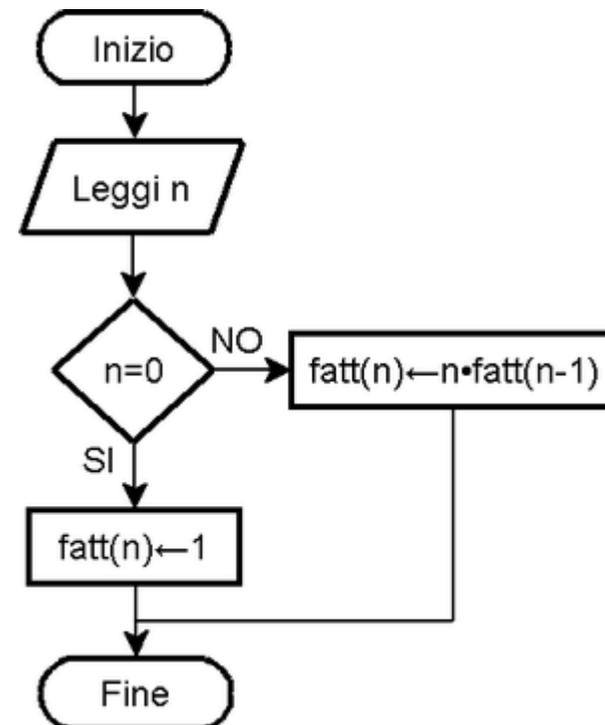
- Linguaggio naturale
- Diagramma di flusso
- Pseudo codice
- Linguaggio di programmazione

Rappresentazione degli algoritmi

- Linguaggio Naturale

- Sollevare il ricevitore
- Attendere il segnale di linea libera
- Comporre il numero
- ...

- Diagramma di flusso



Rappresentazione degli algoritmi

- Pseudo Codice

leggi alfa, beta

prod \leftarrow 0

finché alfa \neq 0 ripeti

prod \leftarrow **prod** + **beta**;

alfa \leftarrow **alfa** - 1;

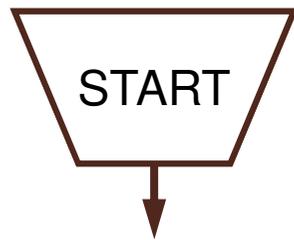
stampa prod;

- Linguaggio di programmazione

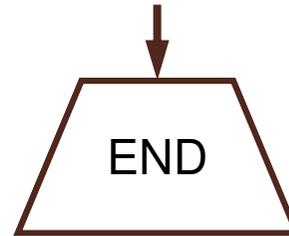
```
#include <stdio.h>
```

```
int main (void) {  
    puts ("ciao");  
    return Exit_success;  
}
```

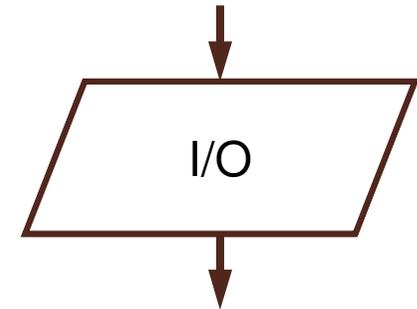
Diagrammi di Flusso



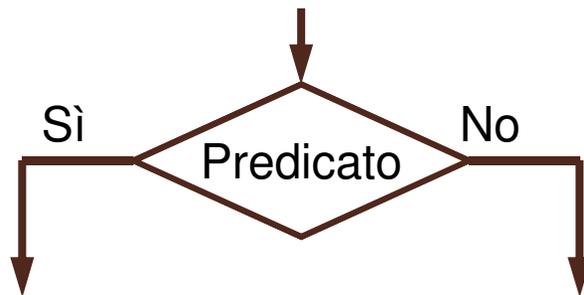
Inizio



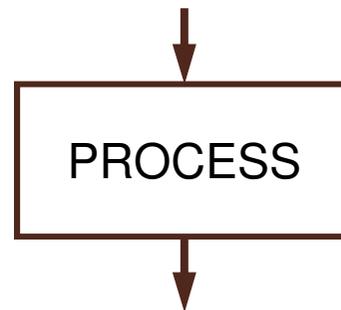
Fine



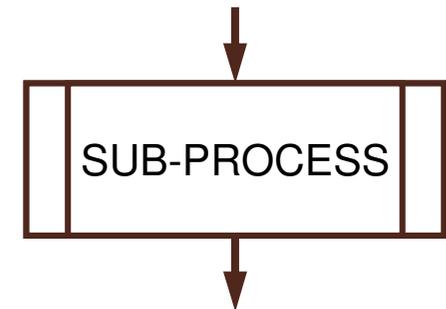
Operazioni di
ingresso/uscita



Selezione a due vie



Elaborazione



Sottoprogramma

Problema da svolgere

Dati in ingresso due numeri X e Y , si calcoli e si visualizzi il numero maggiore.

Algoritmo:

- P1. Leggere un valore in input e assegnarlo alla variabile X
- P2. Leggere un secondo valore in input e assegnarlo alla variabile Y
- P3. Calcolare la differenza tra X e Y
- P4. La differenza è maggiore di zero allora vai al passo 5 altrimenti vai al passo 6
- P5. Visualizza "Il numero maggiore è " X
- P6. Visualizza "il numero maggiore è" Y
- P7. Fine del programma

Soluzione con diagramma di flusso

