

Esercizi sugli algoritmi

Algoritmi

Ricordare che...

- Qualsiasi azione complessa può essere scomposta in una *sequenza di azioni elementari*.
- La sequenza di passi che deriva dall'analisi di un problema viene chiamata **algoritmo** se: 1) è **ordinata**; 2) descrive il problema in maniera **generale**; 3) è **convergente** (cioè ha un inizio e una fine)

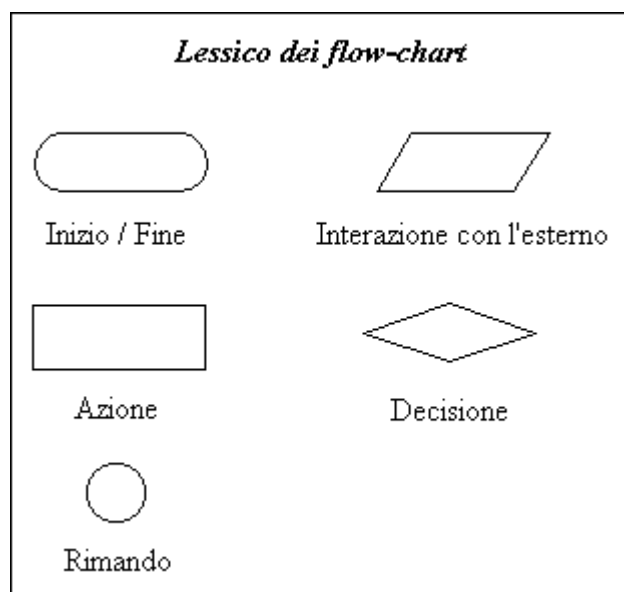
Descrivere le seguenti azioni in maniera algoritmica:

1. Trovare in un mazzo di chiavi quella che apre un lucchetto
2. Fare una telefonata
3. Preparare la cartella per la scuola
4. Calcolare la somma di due numeri
5. Dati 2 numeri in input dare in output il max
6. Dato in input i secondi dare in output minuti, ore e sec
7. Dato in input 10 valori dare in output la somma

Rappresentazione di algoritmi e strutture fondamentali

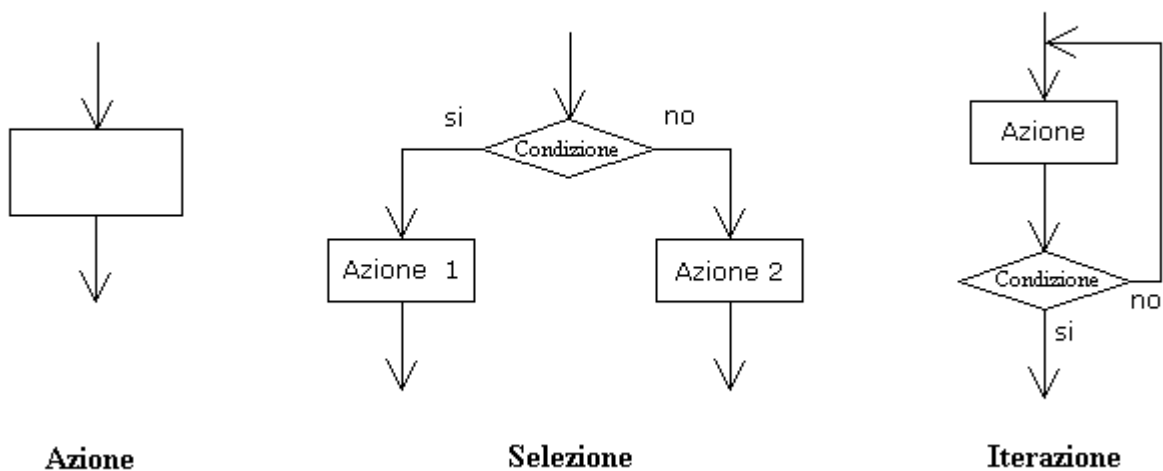
Ricordare che...

- Gli algoritmi possono essere descritti utilizzando vari metodi; alcuni tra i più semplici sono: lista lineare a passi numerati oppure il **diagramma di flusso** (*flow-chart*).
- Il diagramma di flusso è una modalità di rappresentazione *grafica* degli algoritmi; ogni passo è racchiuso in una *box* ed è collegato al successivo tramite una freccia. La forma della *box* definisce il tipo di azione elementare da svolgere (che da ora in poi chiameremo **istruzione**).
- I simboli grafici fondamentali, ovvero i tipi diversi di istruzione sono, in realtà pochissimi: ne possiamo individuare 5 (v. figura)



- Il *flow-chart*, nonostante la sua semplicità e concisione, è un metodo di rappresentazione formale; ciò significa che, anche in presenza di algoritmi banali, va tutto descritto con la massima precisione e in maniera assolutamente univoca (cioè non deve esistere la possibilità di equivocare il significato delle descrizioni).
- Per rendere ancora più rigorosa la rappresentazione degli algoritmi sono state definite le **strutture fondamentali** della programmazione che sono: **azione**, **selezione**, **iterazione**; è stato dimostrato matematicamente che tramite queste sole 3 strutture è possibile descrivere qualsiasi algoritmo. Con la selezione è possibile scegliere l'esecuzione di un'istruzione piuttosto che un'altra in dipendenza da una **condizione logica**. Con un'iterazione possiamo ripetere un gruppo di istruzioni finché non si verifica una condizione logica (che viene chiamata **condizione di fine ciclo**)

Strutture fondamentali



- Una **variabile**, indicata con un simbolo (ad es. sono simboli validi **a**, **cont**, **indice**), è un contenitore per valori di tipo diverso (ad es. numeri). A una variabile può essere attribuito un valore tramite l'istruzione di **assegnazione** (che, nei *flow-chart*, si indica con \leftarrow)
- Possiamo definire alcune variabili con funzioni specifiche; ad es., un **accumulatore** è una variabile che si usa in questa maniera: $a \leftarrow a + b$ in cui a rappresenta l'accumulatore e b è un numero qualsiasi: il significato di questa istruzione è che ad a sarà assegnato il valore che è contenuto in a in quel momento sommato al valore contenuto nella variabile b . Un **contatore** è un caso particolare dell'accumulatore in cui b è uguale a 1. Accumulatori e contatori, di norma, vanno **inizializzati**: prima di usarli all'interno di una struttura iterativa bisogna assegnare ad essi un valore di partenza.

Esercizi (Algoritmi)

- **Provare a descrivere tramite il *flow-chart* i semplici algoritmi degli esercizi precedenti**