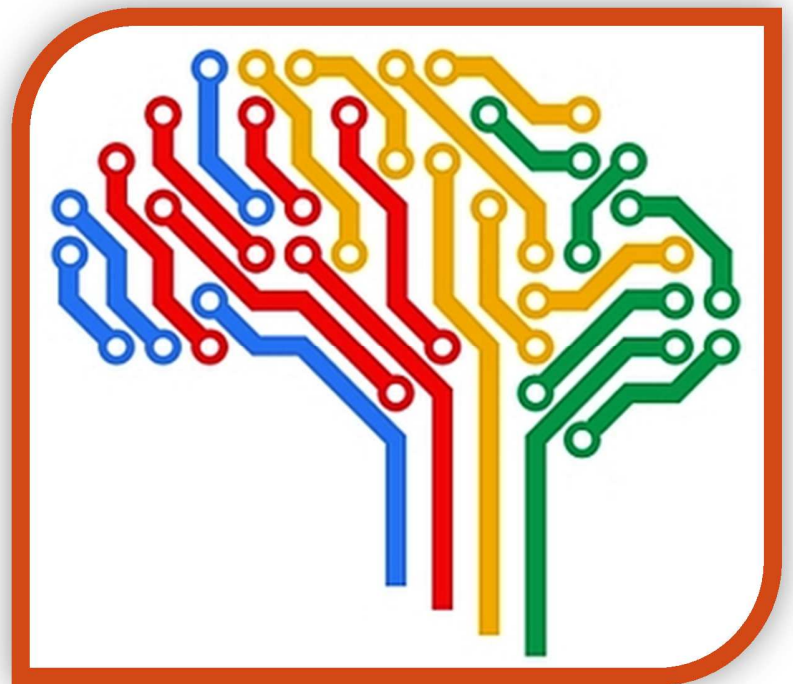


Pensiero computazionale



Antonella Carbonaro

Dipartimento di Informatica - Scienza e Ingegneria

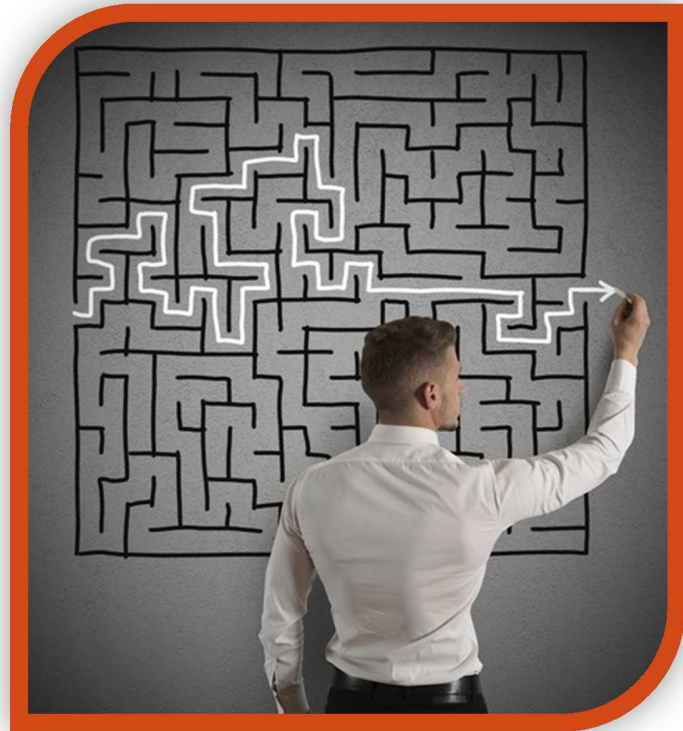
Alma Mater Studiorum, Università di Bologna

27 aprile 2017, Cervia

antonella.carbonaro@unibo.it

Problem solving, competenza chiave del lavoro futuro

Il mercato del lavoro odierno richiede lavoratori in grado di risolvere **problemi non routinari** (World Economic Forum 2016). Secondo il Rapporto sulle competenze degli adulti (Survey of Adult Skill PIACC) le persone incontrano oggi difficoltà crescenti nell'ambito del loro lavoro che presenta situazioni nelle quali è sempre più necessario pensare prima di agire. Una possibile spiegazione della transizione da compiti routinari a compiti non routinari sul posto di lavoro è che, a causa dell'introduzione in grande numero di apparati computerizzati, i lavoratori non devono più eseguire routine manuali. Ad essi è invece richiesto di **affrontare problemi inaspettati e non familiari** nel governare le macchine che gestiscono. Recenti evidenze esaminate nella valutazione di problem-solving degli studenti di 70 Paesi (PISA 2012 Results: Creative Problem Solving) hanno confermato che i risultati sono associati al successo accademico e non alle capacità di ragionamento o all'intelligenza. **Troppo spesso gli insegnanti scoprono che, mentre i loro studenti eccellono in esercizi di routine, falliscono nel risolvere problemi che non hanno mai incontrato prima.** Le competenze di problem-solving sono una componente essenziale delle capacità richieste per **affrontare analiticamente compiti interpersonali e non routinari**. In entrambi i casi le persone devono pensare a come affrontare la situazione, monitorare sistematicamente l'effetto delle loro azioni, e modificarle opportunamente.



Pensiero computazionale: pensiero critico nel valutare le situazioni e una capacità avanzata di problem-solving

Pensiero computazionale: un concetto che sta registrando un interesse crescente, nella scuola e non solo.

Il **pensiero computazionale** è una abilità che andrebbe sviluppata sin da bambini, **a scuola**, perché aiuta a pensare meglio, in modo originale e mai ripetitivo.

Il pensiero computazionale è costituito dai processi di pensiero coinvolti nella formulazione di problemi e delle loro soluzioni, in modo tale che le soluzioni siano rappresentate in una forma che possa essere attuata tramite un sistema di elaborazione dati.



Piano Nazionale Scuola Digitale

Con il Piano Nazionale Scuola Digitale l'insegnamento del pensiero computazionale diventa parte dei programmi della scuola e in particolare di quella primaria.

AZIONE #17

Portare il pensiero logico-computazionale a tutta la scuola primaria



È fondamentale partire dai giovanissimi, per almeno due ragioni: **primo**, anticipare la comprensione della logica sottostante i processi computazionali dell'informatica e delle tecnologie, proprio perché l'avvicinamento alle tecnologie stesse avviene prima, a partire dal contesto familiare; **secondo**, preparare da subito gli studenti allo sviluppo delle competenze che sono al centro del nostro tempo, e saranno al centro delle loro vite e carriere.

L'acquisizione delle competenze relative agli aspetti scientifico-culturali dell'informatica **a prescindere da qualunque aspetto strumentale o tecnologico** aiuta a sviluppare competenze logiche e capacità di risolvere problemi in modo creativo ed efficiente, qualità che sono importanti per tutti i futuri cittadini.

Il pensiero computazionale è utile per qualsiasi professione si svolga o si desideri svolgere perché sviluppa abilità concettuali efficaci nel trattare problemi complessi; ad esempio:

- Incrementa la creatività
- È costruttivo e produce risultati visibili
- Aiuta a padroneggiare la complessità
- Sviluppa il ragionamento accurato e preciso

Pensiero computazionale, cos'è che lo rende così universale. Pensare in modo computazionale significa suddividere il processo decisionale in singoli step, ragionare passo passo sul modo migliore per ottenere un obiettivo. Una comportamento che in realtà – quasi senza accorgercene – mettiamo in atto tutti i giorni, per esempio quando stabiliamo il percorso più breve per raggiungere una destinazione.



ESERCIZI PER PRIMARIA

ESERCIZIO 1

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, problema ricorrente REGOLE E DEDUZIONI.

PROBLEMA

Siano date le seguenti regole:

regola(1,[b,c],a) regola(2,[c,d],a) regola(3,[b,c,d],a) regola(4,[b,a],f)

Trovare:

1. la sigla N della regola che consente di dedurre **a** da **d** e **c**;
2. la lista L che rappresenta il procedimento per dedurre **f** da **b** e **c**.

Scrivere le soluzioni nella seguente tabella.

N	
L	

ESERCIZIO 2

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, problema ricorrente PIANIFICAZIONE.

PROBLEMA

La tabella che segue descrive le attività di un progetto (indicate rispettivamente con le sigle A1, A2, ...), riportando per ciascuna di esse il numero di persone assegnato e il numero di giorni necessari per completarla.

ATTIVITÀ	PERSONE	GIORNI
A1	6	2
A2	4	2
A3	3	3
A4	6	2
A5	4	2
A6	5	1

Le priorità tra le attività sono:

[A1,A2], [A1,A3], [A2,A4], [A4,A6], [A3,A5], [A5,A6].

N.B. Questo significa, per esempio, che la attività A2 può svolgersi solo quando la attività A1 è terminata, o la A6 può svolgersi solo quando la A4 è terminata.

Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, trovare il numero massimo PM di persone che lavorano contemporaneamente al progetto.

N	
PM	

ESERCIZIO 3

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, ELEMENTI DI PSEUDOLINGUAGGIO.

PROBLEMA

Si consideri la seguente procedura ALFA.

```

procedura ALFA;
variables A, B, C, D, E, F integer;
input A, B;
C ← A + B;
D ← A × B;
E ← C + D;
F ← (A + 4) × (A - B);
output C, D, E, F;
endprocedura;

```

I valori in input sono: 5 per A, 1 per B; determinare i valori di output di C, D, E, F e scriverli nella seguente tabella.

C	
D	
E	
F	



ESERCIZI PER SECONDARIA I GRADO

ESERCIZIO 1

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, problema ricorrente *KNAPSACK*.

PROBLEMA

In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto da un termine che contiene le seguenti informazioni.

minerale(<sigla del minerale>, <valore in euro>, <peso in Kg>).

Il deposito contiene i seguenti minerali:

minerale(m1,80,43)	minerale (m2,60,34)	minerale (m3,65,32)
minerale(m4,60,34)	minerale (m5,60,36)	minerale (m6,83,51)

Disponendo di un motocarro con portata massima di 100 Kg, trovare la lista L delle sigle di tre minerali diversi che siano trasportabili contemporaneamente con questo mezzo e che abbiano il massimo valore complessivo.

N.B. Nelle liste, elencare le sigle in ordine (lessicale) crescente; per le sigle usate si ha il seguente ordine: $m_1 < m_2 < m_3 < \dots$.

L	
---	--

ESERCIZIO 2

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, problema ricorrente PERCORSI IN UN GRAFO.

PROBLEMA

Un grafo (che si può pensare corrispondere alla rete di strade che collegano delle città) è descritto dal seguente elenco di archi:

arco(n1,n2,11)	arco(n2,n3,5)	arco(n3,n4,10)	arco(n2,n4,4)
arco (n4,n5,3)	arco(n5,n1,2)	arco(n4,n1,7)	arco(n1,n6,6)

Disegnare il grafo e trovare:

1. la lista L1 del percorso semplice più breve tra n1 e n3;
2. la lista L2 del percorso semplice più lungo tra n1 e n3.

L1	
L2	

ESERCIZIO 3

Si faccia riferimento alla GUIDA ALLA SOLUZIONE DEI PROBLEMI – OPS 2017, ELEMENTI DI PSEUDOLINGUAGGIO.

PROBLEMA

Si consideri la seguente procedura BETA.

```

procedure BETA;
variables A, J integer;
A ← 0;
for J from 1 to 4 step 1 do;
    A ← A + J + 1;
endfor;
output A;
endprocedure;

```

Determinare il valore di output di A.

A	
---	--