Co-funded by the
Erasmus+ Programme
of the European Union

Trainers Support

Algorithmic Thinking

Algorithmic Thinking for Migrants Teachers Education

2021-1-EL01-KA210-ADU-000035033

DISPENSA #2

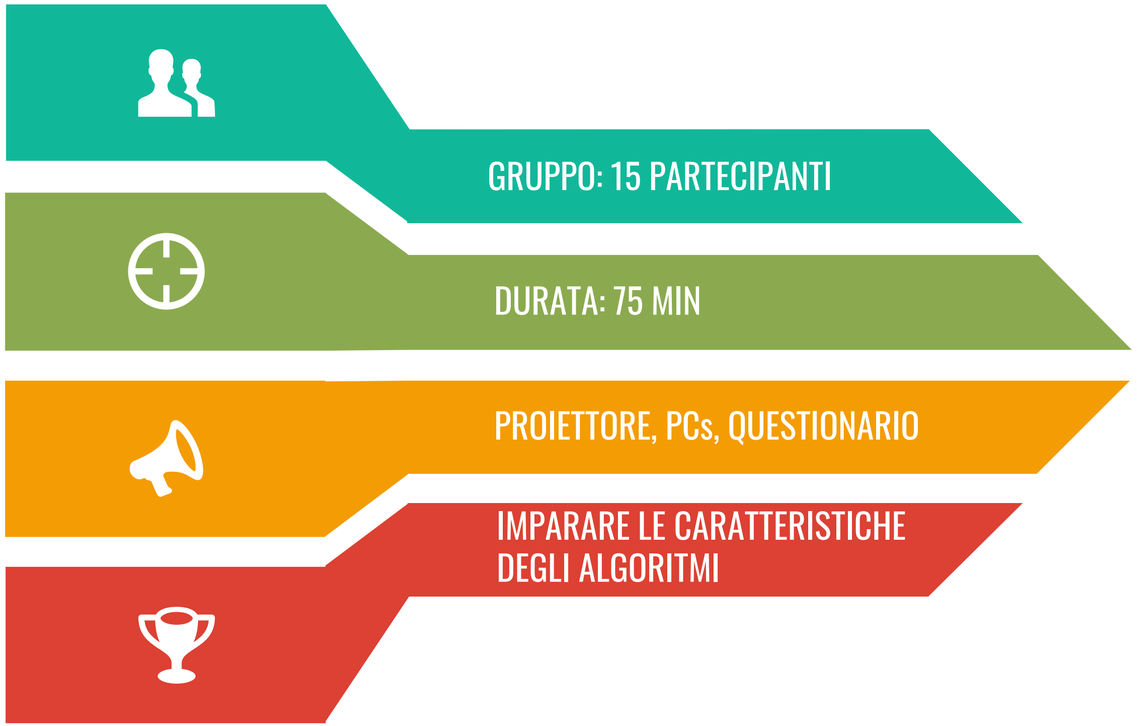
TITOLO: IL PENSIERO ALGORITMICO

Co-funded by the
Erasmus+ Programme
of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

INDICAZIONI PER LA LEZIONE



ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ
IKY
Co-funded by the Erasmus+ Programme of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

COS'UN ALGORITMO???

Un algoritmo è un insieme di procedure passo-passo, o un insieme di regole da seguire, per completare un compito specifico o risolvere un particolare problema. La parola algoritmo è stata coniata per la prima volta nel 9° secolo. Gli algoritmi sono ovunque intorno a noi. Esempi comuni sono: la ricetta per cucinare una torta, il metodo che usiamo per risolvere un problema di divisione lunga e il processo per fare il bucato. Ecco come potrebbe apparire la preparazione di una torta, scritta come un elenco di istruzioni, proprio come un algoritmo:

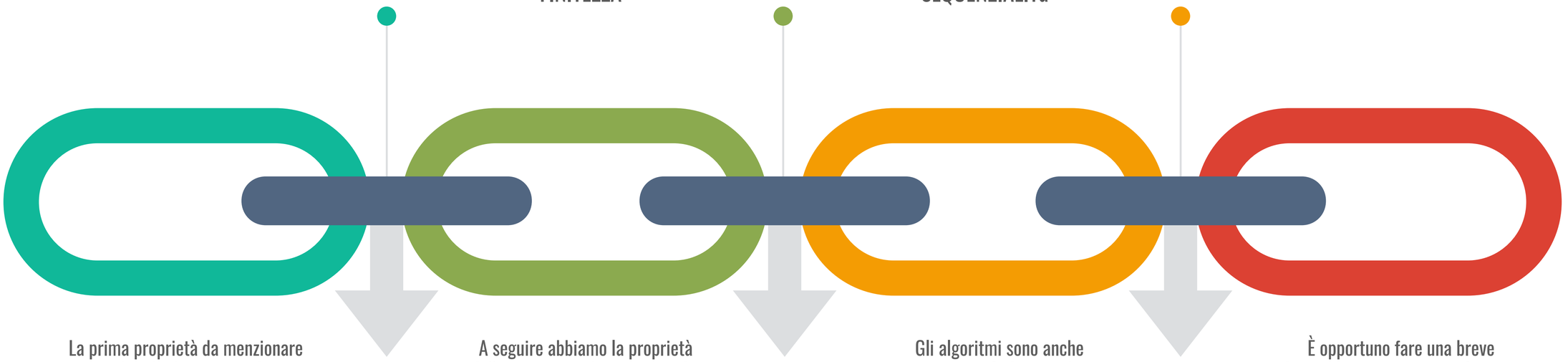
1. Preriscaldare il forno
2. Preparare gli ingredienti
3. Misurare gli ingredienti
4. Mescolare gli ingredienti per ottenere la pastella
5. Ungere una teglia
6. Versare la pastella nella teglia
7. Mettere la teglia nel forno
8. Impostare un timer
9. Quando il timer scatta, estrarre la teglia dal forno

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

LE PROPRIETà DELL'ALGORITMO

INSIEME DI PASSAGGI SPECIFICI



La prima proprietà da menzionare non fa che ribadire quanto detto in precedenza: un algoritmo è **un insieme di singoli passaggi**. Una ricetta si adatta a questa analogia in modo molto semplice, piena com'è di passaggi come: 'preriscaldare il forno a 180 gradi Celsius' o 'aggiungere due cucchiai di zucchero all'impasto'.

FINITEZZA

A seguire abbiamo la proprietà della **finitezza**, che significa che ogni passo deve essere definito con precisione. Ogni passo di un algoritmo può avere uno e un solo significato, altrimenti è ambiguo. Allo stesso modo, i cuochi non scrivono mai cose come "un po' di zucchero" o "cuocere per un po'"

SEQUENZIALITà

Gli algoritmi sono anche **sequenziali**. Le fasi che compongono il processo devono essere eseguite nell'ordine specificato. In caso contrario, il risultato dell'esecuzione dell'algoritmo sarà probabilmente errato. Come in una ricetta, dobbiamo rispettare la sequenza quando eseguiamo un algoritmo perché abbia un risultato significativo.

STATO DELL'ALGORITMO

È opportuno fare una breve deviazione per esaminare il motivo per cui la sequenza è così importante per gli algoritmi. Tutto ha a che fare con lo stato, cioè con i valori attuali di tutte le cose di cui l'algoritmo tiene traccia. La sequenza dei passi di un algoritmo assicura che lo stato cambi sempre nello stesso modo ogni volta che l'algoritmo viene eseguito.

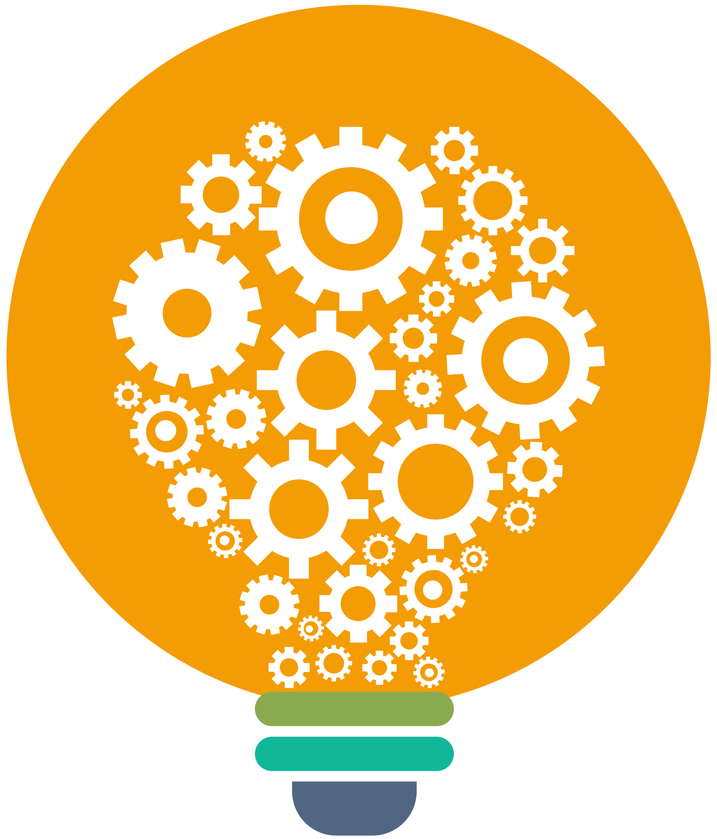
ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

TIPOLOGIE DI ALGORITMI

Gli algoritmi sono classificati in base ai concetti che utilizzano per svolgere un compito. Sebbene esistano molti tipi di algoritmi, quelli fondamentali sono:

- **Algoritmi divide et impera** - dividono il problema in sottoproblemi più piccoli dello stesso tipo; risolvono questi problemi più piccoli e combinano le soluzioni per risolvere il problema originale.
- **Algoritmi di forza bruta** - provano tutte le soluzioni possibili fino a trovare una soluzione soddisfacente.
- **Algoritmi randomizzati** - utilizzano un numero casuale almeno una volta durante il calcolo per trovare una soluzione al problema.
- **Algoritmi greedy** - trovano una soluzione ottimale a livello locale con l'intento di trovare una soluzione ottimale per l'intero problema.
- **Algoritmi ricorsivi** - risolvono la versione più bassa e semplice di un problema per poi risolvere versioni sempre più grandi del problema fino a trovare la soluzione del problema originale.
- **Algoritmi di backtracking** - dividono il problema in sottoproblemi, ognuno dei quali può essere tentato di risolvere; tuttavia, se non si raggiunge la soluzione desiderata, si procede a ritroso nel problema fino a trovare un metodo che lo faccia progredire.



LEZIONE #2 - PENSIERO ALGORITMICO

ALGORITMI DI ORDINAMENTO

Un **algoritmo di ordinamento** è un algoritmo che mette gli elementi di un elenco in un certo ordine. L'ordinamento è spesso un primo passo importante negli algoritmi che risolvono problemi più complessi. Esiste un gran numero di algoritmi di ordinamento, ognuno con i propri vantaggi e costi. Di seguito ci concentreremo su alcuni degli algoritmi di ordinamento più famosi.

- **Ordinamento lineare**: Trova l'elemento più piccolo nell'elenco da ordinare, lo aggiunge a un nuovo elenco e lo rimuove dall'elenco originale. Ripete l'operazione finché l'elenco originale non è vuoto.
- **Ordinamento a bolle**: Confronta i primi due elementi dell'elenco e, se il primo è maggiore del secondo, li scambia. Ripete questa operazione con ogni coppia di elementi adiacenti dell'elenco. Quindi, ripete questo processo finché l'elenco non è completamente ordinato.
- **Ordinamento per inserimento**: Confronta ogni elemento dell'elenco con tutti gli elementi precedenti fino a trovare un elemento più piccolo. Scambia questi due elementi. Ripete questo processo finché l'elenco non è completamente ordinato.

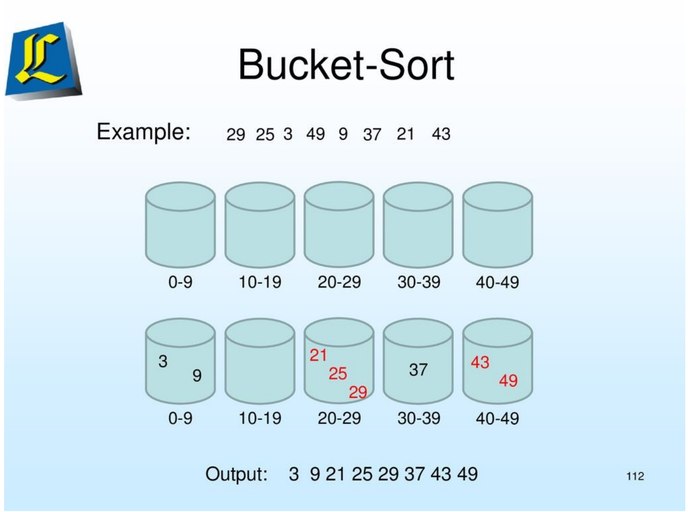
ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

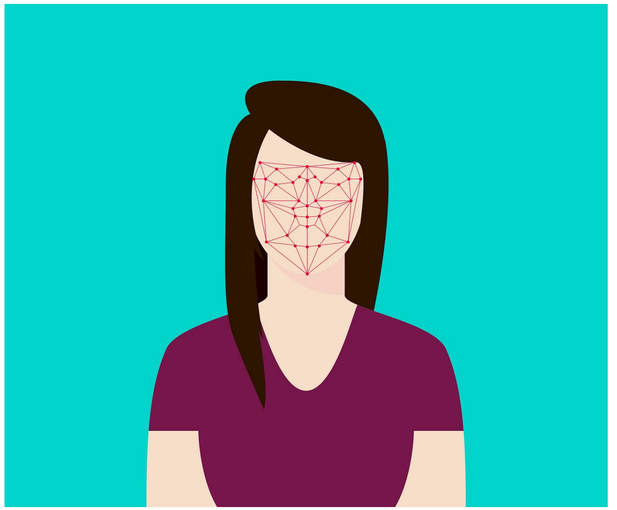
Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

ESEMPI DI ALGORITMI NEL MONDO REALE

RIORDINO DOCUMENTI #1



RICONOSCIMENTO FACCIALE #2



SEMAFORO #4

RICERCA SU GOOGLE #3

ORARI AUTOBUS #5

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LEZIONE #2 - PENSIERO ALGORITMICO

Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

ATTIVITà #2.1

Il formatore distribuirà un foglio con un problema da risolvere. Il problema riguarda un pastore che intende attraversare un fiume utilizzando la sua barca con il seguente carico:

- una pecora
- un lupo
- un po' d"erba

Il trasporto deve essere portato a termine in sicurezza perché il lupo potrebbe mangiare la pecora o la pecora potrebbe mangiare l'erba. La capacità della barca è piccola, quindi può far passare solo un carico di quelli sopra elencati.

I formatori devono progettare l'algoritmo.

La riflessione durerà 10 minuti.

COMPETENZE CHIAVE SVILUPPATE

- Capacità di problem solving
- Capacità di ragionamento
- Competenze ambientali
- Competenze tecniche
- Capacità di gestione del tempo

TEMPO

30 min

MATERIALI RICHIESTI

PC, proiettore, fogli

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

RIFERIMENTI BIBLIOGRAFICI

BEECHER, KARL. 2017. COMPUTATIONAL THINKING: A BEGINNER'S GUIDE TO PROBLEM-SOLVING AND PROGRAMMING. SWINDON, ENGLAND: BCS: THE CHARTERED INSTITUTE FOR IT.

KNUTH, D. (1997) THE ART OF COMPUTER PROGRAMMING, VOLUME 1: FUNDAMENTAL ALGORITHMS. BOSTON, MA, USA: ADDISON-WESLEY

PANE, J. F. ET AL. (2001) STUDYING THE LANGUAGE AND STRUCTURE IN NON-PROGRAMMER'S SOLUTIONS TO PROGRAMMING PROBLEMS. INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES, 54 (2). 237.

PEA, R. ET AL. (1987) THE BUGGY PATH TO THE DEVELOPMENT OF PROGRAMMING EXPERTISE. FOCUS ON LEARNING PROBLEMS IN MATHEMATICS, 9 (1). 5.