

Lo sviluppo dell'algoritmo





Ogni giorno, consapevolmente o meno, utilizziamo degli algoritmi



Dalla preparazione di una ricetta, al risolvere un problema matematico, ci affidiamo a una serie di passaggi logici per raggiungere un risultato



Ma quando si tratta di programmazione, come sviluppiamo un algoritmo?

Definizione di Algoritmo



Prima di tuffarci nell'arte dello sviluppo dell'algoritmo, dobbiamo comprendere cosa sia esattamente un algoritmo



Un algoritmo è una sequenza finita e ben definita di passaggi che, se seguiti correttamente, portano alla soluzione di un problema specifico



Un algoritmo è come una ricetta: ha ingredienti , passaggi da seguire e un risultato finale



Importanza della chiarezza

- Quando sviluppiamo un algoritmo, la chiarezza è essenziale
- Se la nostra 'ricetta' non è chiara, chi la segue potrebbe non ottenere il risultato desiderato
- La chiarezza è fondamentale per evitare errori

Passi per sviluppare un algoritmo



Comprensione del problema: Prima di iniziare a scrivere un algoritmo, è fondamentale comprendere appieno il problema che si intende risolvere



Definizione degli input e output: Dobbiamo sapere quali informazioni avremo a disposizione e cosa vogliamo ottenere al termine dell'esecuzione



Strutturazione dei passaggi: Questo è il cuore dell'algoritmo



Testare l'algoritmo: Una volta scritto l'algoritmo, è fondamentale testarlo per verificare che funzioni come previsto

Esempi di sviluppo di algoritmi



Supponiamo di voler sviluppare un algoritmo che, dato un elenco di numeri, trovi il numero più grande



La sequenza potrebbe essere: iniziare con il primo numero e considerarlo come il più grande finora



Continuare così per tutta la lista. Algoritmi più complessi potrebbero richiedere passaggi aggiuntivi, strumenti matematici o strutture dati particolari, ma la logica di base rimane la stessa: una sequenza di passaggi per ottenere un risultato



Sviluppare algoritmi è una delle competenze fondamentali in informatica e programmazione



La capacità di pensare in modo logico, sequenziale e strutturato, considerando tutte le variabili, è essenziale per creare programmi efficienti e funzionanti



Con pratica e esperienza, diventerete sempre più abili nell'arte dello sviluppo di algoritmi

Conclusione



1. Algoritmo per trovare la somma di due numeri

Problema: Dati due numeri, trova la loro somma.

1. Algoritmo per trovare la somma di due numeri

- **Input:** Due numeri (ad esempio, 5 e 7).
Output: La somma dei due numeri (ad esempio, 12).
- **Passaggi:**
 1. Prendi il primo numero.
 2. Aggiungi al primo numero il valore del secondo numero.
 3. Restituisci il risultato.

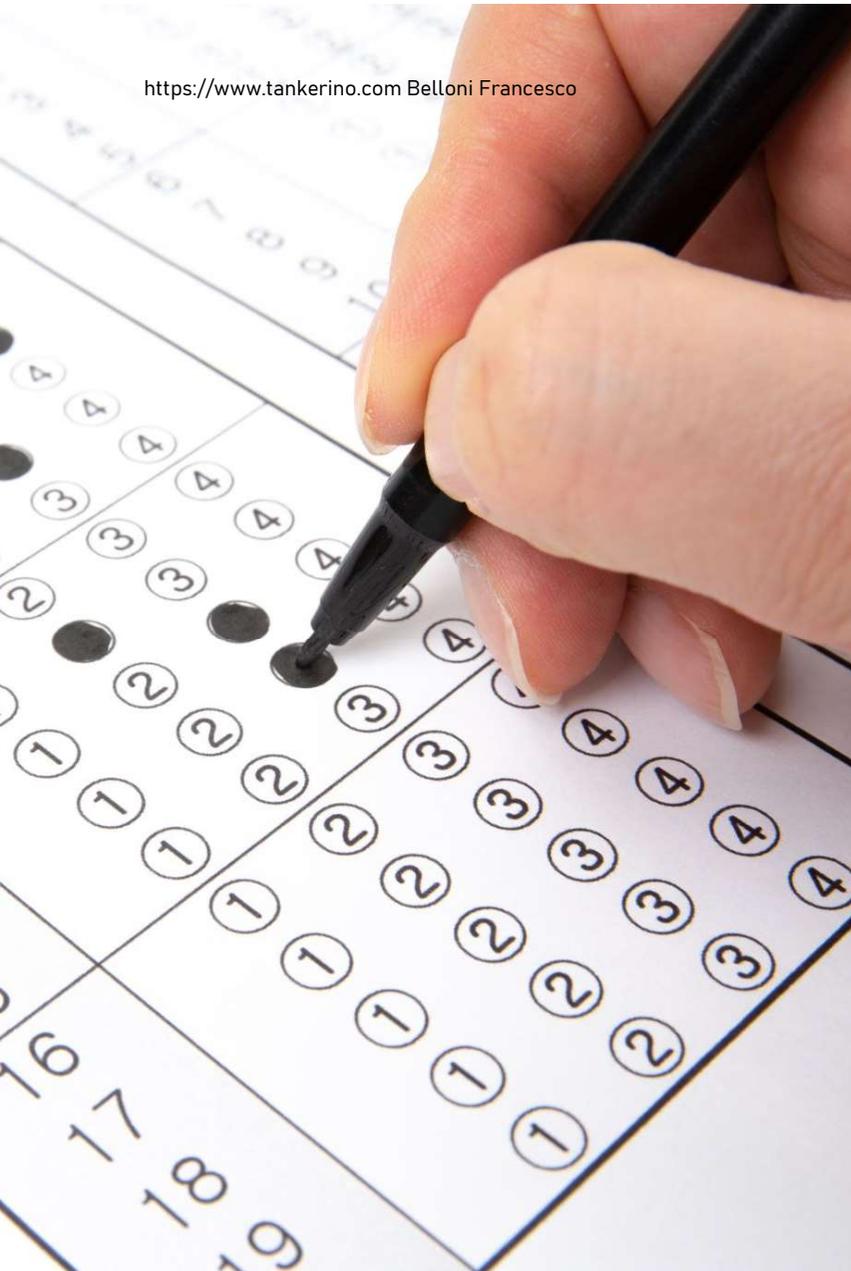


2. Algoritmo per trovare il numero più grande in una lista

Problema: Dato un elenco di numeri, determina il numero più grande.

2. Algoritmo per trovare il numero più grande in una lista

- **Input:** Una lista di numeri (ad esempio, [3, 8, 2, 7, 5]).
- **Output:** Il numero più grande (in questo caso, 8).
- **Passaggi:**
 1. Considera il primo numero della lista come il più grande.
 2. Confronta il numero più grande corrente con il numero successivo nella lista.
 3. Se il numero successivo è più grande del numero più grande corrente, assegnalo come nuovo numero più grande.
 4. Ripeti il passo 2 e 3 fino alla fine della lista.
 5. Restituisci il numero più grande.



3. Algoritmo per trovare tutti i numeri pari in una lista

- Problema: Dato un elenco di numeri, trova tutti i numeri pari

3. Algoritmo per trovare tutti i numeri pari in una lista

Input: Una lista di numeri (ad esempio, [4, 7, 8, 10, 13]).

Output: Una lista di numeri pari (in questo caso, [4, 8, 10]).

▪ **Passaggi:**

1. Inizia con una lista vuota per i numeri pari.
2. Esamina ogni numero della lista.
3. Se il numero è divisibile per 2 (cioè, il resto della divisione per 2 è 0), aggiungilo alla lista dei numeri pari.
4. Continua finché non hai esaminato tutti i numeri della lista.
5. Restituisci la lista dei numeri pari.



4. Algoritmo per invertire le parole in una frase

- Problema: Dato una frase, inverti l'ordine delle parole



4. Algoritmo per invertire le parole in una frase

- **Problema:** Dato una frase, inverti l'ordine delle parole.
Input: Una frase (ad esempio, "Oggi è una bella giornata").
Output: Frase con le parole invertite (in questo caso, "giornata bella una è Oggi").
- **Passaggi:**
 1. Dividi la frase in parole, utilizzando lo spazio come separatore.
 2. Inizia con una nuova frase vuota.
 3. Prendi l'ultima parola della frase originale e aggiungila alla nuova frase.
 4. Continua ad aggiungere parole dalla fine all'inizio.
 5. Quando tutte le parole sono state aggiunte, restituisci la nuova frase.

5. Algoritmo di ordinamento bubble sort

- Problema: Ordina un elenco di numeri in ordine crescente utilizzando il metodo bubble sort

5. Algoritmo di ordinamento bubble sort

- **Problema:** Ordina un elenco di numeri in ordine crescente utilizzando il metodo bubble sort.
Input: Una lista di numeri (ad esempio, [64, 34, 25, 12, 22, 11, 90]).
Output: La lista ordinata (in questo caso, [11, 12, 22, 25, 34, 64, 90]).
- **Passaggi:**
 1. Confronta ogni coppia di numeri adiacenti nella lista.
 2. Se una coppia non è nell'ordine corretto (il numero più grande precede il numero più piccolo), scambiali.
 3. Continua attraverso l'intera lista fino a quando non sono stati fatti scambi in un passaggio completo attraverso la lista.
 4. Se non sono stati effettuati scambi in un passaggio completo, l'elenco è ordinato e l'algoritmo è completo. Altrimenti, ripeti dal passo 1.
- Spero che questi esempi ti aiutino a comprendere meglio il concetto di algoritmo e la sua applicazione pratica.