

Prefazione

Questo libro è basato su corsi che sono stati offerti al Massachusetts Institute of Technology (MIT) dal 2006 e come “Massive Online Open Courses” (MOOC) attraverso edX e MITx dal 2012 in poi. La prima edizione si basava su un corso singolo della durata di un semestre, ma con il tempo non sono riuscito a fare a meno di aggiungere altro materiale che poteva trovare posto in un semestre. Questa edizione è adatta per un corso introduttivo di *computer science* distribuito su due semestri o tre trimestri.

Il libro si rivolge a lettori con poca o nessuna esperienza di programmazione, che desiderano comprendere gli approcci computazionali alla risoluzione di problemi.

Mettiamo l'accento qui più sull'ampiezza che sulla profondità. L'obiettivo è dare ai lettori una breve introduzione a molti temi, perché possano avere un'idea di ciò che è possibile quando verrà il momento di pensare come usare la computazione per raggiungere un obiettivo. Detto ciò, questo non è un libro di superficie “cultura informatica”: è impegnativo e rigoroso. I lettori che vogliono davvero apprendere questo materiale dovranno dedicare un bel po' di tempo e fare un po' di fatica per imparare come piegare il computer alla loro volontà.

Lo scopo principale del libro è aiutare i lettori ad acquisire le competenze per un uso produttivo delle tecniche computazionali. Dovranno imparare ad usare modi di pensare computazionali per inquadrare i problemi, costruire modelli computazionali e guidare il processo di estrazione di informazioni dai dati. La conoscenza principale che ricaveranno da questo libro è l'arte del *problem solving* computazionale.

Abbiamo scelto di non includere dei problemi alla fine dei capitoli; abbiamo invece inserito degli “esercizi” in punti opportuni all'interno dei capitoli. Alcuni sono molto brevi e mirano a dare ai lettori la conferma di avere compreso il materiale letto; altri sono più impegnativi e sono adatti anche come domande d'esame; altri ancora sono abbastanza impegnativi da poter essere utili come “compiti a casa”.

I capitoli 1-13 contengono il tipo di materiale che di solito fa parte di un corso introduttivo di informatica, ma la presentazione non è convenzionale. Intrecciamo quattro filoni di materiali:

- gli elementi base della programmazione;
- il linguaggio di programmazione Python 3;
- tecniche computazionali di risoluzione dei problemi;
- la complessità computazionale.

Copriamo la maggior parte delle funzionalità di Python, ma l'accento è su quello che si può fare con un linguaggio di programmazione, non sul linguaggio stesso. Per esempio, alla fine del Capitolo 3, avremo visto solo una piccola parte di Python ma avremo già introdotto le nozioni di enumerazione esaustiva, algoritmi “ipotizza e verifica”, ricerca per bisezione e algoritmi di approssimazione efficienti. Introduciamo le varie funzionalità di Python progressivamente nel corso del libro; analogamente, introduciamo vari aspetti dei metodi di programmazione lungo tutto l'arco del testo. L'idea è aiutare i lettori a imparare Python e come essere buoni programmatori nel contesto dell'uso della computazione per risolvere problemi interessanti. Questi capitoli sono stati riveduti in modo da rendere più graduale l'apprendimento e per inserire un maggior numero di esercizi rispetto alla seconda edizione del libro.

Il Capitolo 13 contiene un'introduzione alla creazione di grafici in Python, un argomento che spesso non viene preso in considerazione nei corsi introduttivi; noi però siamo convinti che apprendere a produrre visualizzazioni delle informazioni sia importante e che questo tema debba essere incluso in un corso introduttivo di informatica. Il capitolo comprende materiali non affrontati nella seconda edizione.

I Capitoli 14-17 riguardano l'uso della computazione come mezzo per capire il mondo reale. Coprono temi che pensiamo debbano diventare parte di un normale secondo corso in un curriculum di informatica. Non viene presupposta alcuna conoscenza della matematica al di là dell'algebra delle scuole superiori, ma diamo per scontato che il lettore si trovi a proprio agio con un modo di ragionare rigoroso e che non si spaventi davanti ai concetti matematici.

Questa parte del libro è dedicata ad argomenti che non trovano spazio nella maggior parte dei testi introduttivi: visualizzazione e analisi dei dati e apprendimento automatico. Fatta eccezione per il Capitolo 14, il materiale in questa sezione del libro si concentra sugli aspetti concettuali più che sulla programmazione. Il Capitolo 14 è un'introduzione a Pandas, argomento che non era coperto nelle edizioni precedenti.

Tre sono i temi portanti che caratterizzano tutto il libro: la risoluzione di problemi in modo sistematico, il potere dell'astrazione e la computazione come strumento per pensare il mondo. Quando avrete finito questo libro, avrete raggiunto diversi risultati:

- avrete imparato un linguaggio, Python, per esprimere computazioni;
- avrete imparato un approccio sistematico all'organizzazione, alla scrittura e alla messa a punto di programmi di medie dimensioni;
- avrete sviluppato una comprensione informale della complessità computazionale;
- avrete sviluppato qualche idea del processo da seguire per passare dall'enunciazione ambigua di un problema alla formulazione computazionale di un metodo per risolvere quel problema;

- avrete imparato un insieme utile di tecniche algoritmiche e di riduzione dei problemi;
- avrete imparato a utilizzare strumenti computazionali (fra cui semplici strumenti statistici, di visualizzazione e di apprendimento automatico) per creare modelli e comprendere dati.

Programmare è un'attività intrinsecamente difficile. Come “non c'è una via regale alla geometria”, non ci sono vie regali nemmeno nella programmazione. Se davvero volete imparare questo materiale, leggere il libro non sarà sufficiente: come minimo dovrete completare gli esercizi che comportano la scrittura di codice. Se vi sentite pronti per affrontare qualche compito più ambizioso, provate qualcuno dei problemi che si trovano agli URL

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>

e

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/>

1. Questa fu la presunta risposta di Euclide, intorno al 300 a.C., alla richiesta del re Tolomeo di un modo più semplice per imparare la matematica.