Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

20 aprile 2026

Nella lezione di oggi abbiamo implementato classi e metodi che abbiamo utilizzato per fare il *parsing* delle transazioni e per identificare le componenti da cui sono formate. Gli esercizi di questa nota si riferiscono al codice scritto in aula, che potete scaricare qui: `https://www.mat.uniroma2.it/~pasquale/dida/aa2526/pcd/pcd260420.zip`

Esercizio 1. Nel file `transaction.py`, per ognuna delle classi `Tx`, `TxIn` e `TxOut` abbiamo implementato un metodo di classe `parse` che prende in input una sequenza di byte opportuna e restituisce l'oggetto codificato nella sequenza.

Per ognuna delle classi, scrivere un metodo `serialize` che restituisca una sequenza di byte contenente la serializzazione dell'oggetto della classe.

Esercizio 2. Scrivere un metodo `hash` per la classe `Tx` che restituisca l'hash256 della serializzazione della classe.

Esercizio 3. Scrivere un metodo `tot_out` per la classe `Tx` che restituisca la somma degli `amount` contenuti negli output della transazione.

Esercizio 4. Scrivere un metodo `fee` per la classe `Tx` che restituisca la *transaction fee* della transazione. La transaction fee è la differenza fra la somma degli `amount` contenuti negli output puntati dagli input della transazione e la somma degli `amount` contenuti negli output della transazione. Per recuperare gli output puntati dagli input di una transazione potete usare, per esempio, le api di un qualche sito che consente di recuperare dati dalla Blockchain, per esempio `https://mempool.space/docs/api/rest#get-transaction-hex`.

DRAFT