區塊鏈模擬器完整說明文件

一、Design & Algorithm Description

1. 區塊鏈結構（Blockchain Structure）

本模擬器的區塊鏈結構以列表形式儲存區塊，每個區塊（`Block`）包含：

- `index`：區塊在區塊鏈中的位置編號。
- `previous_hash`：前一區塊的雜湊值，用來確保鏈的連續性與不可篡改性。
- `transactions`：區塊內包含的交易清單。
- `timestamp`：區塊建立時間（UNIX timestamp）。
- `nonce`：挖礦過程中的變數，用於改變雜湊值以達成指定難度（Proof of Work）。
- `hash`：經 SHA-256 雜湊後的區塊唯一值。

第一個區塊為創世區塊（Genesis Block），其 `previous_hash` 為 64 個零。

2. 交易格式與驗證機制（Transaction Format & Validation）

每筆交易（`Transaction`）包含：

- `sender`：發送者名稱
- `receiver`：接收者名稱
- `value`：交易金額（正整數）
- `timestamp`：交易建立時間
- `signature`：交易的數位簽章（以 RSA 私鑰簽名）

驗證步驟：

1. 確保金額為正整數且發送者與接收者不同。
2. 確保發送者餘額足夠。
3. 使用 RSA 公鑰驗證簽章。

3. 挖礦與區塊建立邏輯（Mining & Block Creation）

- 當交易池（transaction pool）超過 5 筆或超過 60 秒未新增區塊時，啟動挖礦。

- 系統會挑選 IP 位址數值最小的參與者作為礦工。
- 礦工透過調整 `nonce` 嘗試產生一個 SHA-256 雜湊值，以符合「前綴為兩個零（"00"）」的難度要求。
- 挖礦成功後將該區塊加入鏈中，並清空已打包交易。

4. 共識機制（Consensus Mechanism）

此模擬器採單一執行個體，無分散式節點，因此共識機制為簡化版：

- 選擇 IP 值最小的參與者作為礦工。
- 沒有區塊衝突處理（無分叉解決策略）。
- 無需多方驗證；驗證僅在單機內進行。

二、Documentation

1. 如何執行模擬器（How to Run the Simulation）

方法一：使用 Python 本地執行

環境需求：

- Python 3.8 或以上版本
- 安裝依賴套件：

```
pip install -r requirements.txt
```

執行指令：

```
python blockchain_sim.py
```

參數：

- `-debug`：顯示詳細除錯日誌
- `-parties N`：設定參與者人數（預設為 5）

方法二：使用 Docker 執行

步驟：

```
cd blockchain_sim

docker build -t blockchain-sim .
docker run blockchain-sim              # 基本執行

docker run blockchain-sim --debug      # 啟用除錯模式
docker run blockchain-sim --parties 3  # 設定參與者人數
docker run blockchain-sim --debug --parties 3  # 多參數組合
```

2. 如何驗證區塊鏈（How to Verify the Blockchain）

系統會定期自動驗證整個鏈的有效性（約每十次操作隨機驗證一次）及最後模擬結束時的完整驗證。

驗證內容包括：

- 創世區塊正確性
- 區塊索引與 `previous_hash` 一致性
- 每筆交易經簽章驗證且合法
- 區塊雜湊符合難度要求

3. Docker 腳本與問題排除（Docker Images and Troubleshooting）

Dockerfile 範例：

```
FROM python:3.9-slim
WORKDIR /app
COPY blockchain_sim.py .
COPY requirements.txt .
RUN pip install -r requirements.txt
CMD ["python", "blockchain_sim.py"]
```

常見問題排除：

- 權限錯誤：使用 `sudo` 執行 Docker
- 映像無法建立：確認 Dockerfile 存在且語法正確
- 無法運行：檢查是否已安裝 Docker 並確認服務已啟動

- 日誌查看：

```
docker logs blockchain-sim
```

環境清理：

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker rmi blockchain-sim
```

三、總結

本模擬器提供簡化但完整的區塊鏈模擬環境，透過 Python 與 Docker 容器技術，實現交易建立、區塊打包、挖礦驗證、鏈結驗證等核心功能。

其設計目的是實驗用途，幫助使用者理解：

- 交易簽章與驗證流程
- 區塊鏈鏈結性與不可竄改性
- 挖礦與共識模擬

並具備完整操作指引，使用者可輕鬆模擬與調整參數以觀察行為變化。