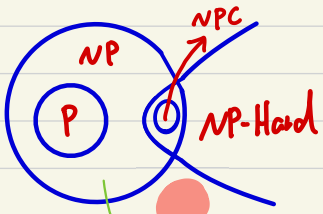
當交易發生至進入區塊鏈需要一點時間

Q: problem a: answer

M(Q) → a

NP def: Given "a" for Q , Time (verify a of Q) is polynomial time

P def: find "a" in poly



若題目設計在這，無法驗證，不適合做題目

適合在這設計題目

Proof-of-Work : 耗電量極大

Proof-of-Stack : 全連 信卡住

雜湊 ≠ 加密

if k=4 bit

input : 0001 0000 → 0001 0000 ↘

input : 0001 00 00 → 0001 00 00 ↗ 不同輸入結果相同了，所以需 length padding 做一補充使結果不同

	length
0001 0000	1000
0001 0000	0110

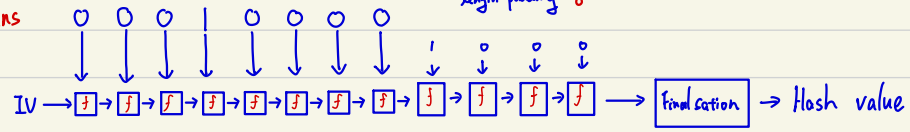
長度不夠除 k，所以剩餘補 0。

除了用字串長度做為 length padding，也可用補 0 的個數

Quiz :  , Hash data : 00010000 , 求過程

length padding 8

Ans



Puzzle

$SHA256 : \{0,1\}^{*} \to \{0,1\}^{256}$

Hash (Prev-Hash , Pack , Nounce)

256 bit, 256bit, 256 bit

Pack tx_1 tx_2 tx_3 tx_4 .... → Pack 256bit

Hash ($tx_1, tx_2, \dots, tx_{5000}$)

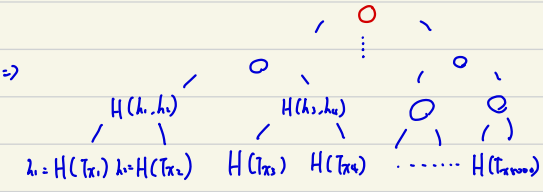
tx_1 → □ → .... → tx_{5000} → length → □ → 256 bit output ⇒ 它好慢!! dependence，所以不用這種方式

Merkle Tree (Hash Tree)

$H(Tx_1)$ $H(Tx_2)$ $H(Tx_3)$ .... $H(Tx_{5000})$ 平行處理，但沒有減低計算量，且輸出 5000 x 256bit 有空間問題

$H(Tx_1, Tx_2, \dots, Tx_{5000})$ 串列處理，慢 時間 n，輸出 256 bit

Ex W_k ⇒

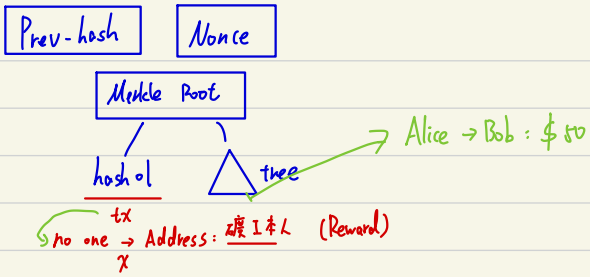


所以又需要記 0 256 bit，但計算數 x 2，但若平行運算的話就是一層一層算，則時間就變成 $\log n$

還有一優點，不可逆回 Origin data

Summary : Parallel computation on Merkle Tree : Deal with each layer, Time $O(\log n)$

on arbitrary input length hash : Time : $O(n)$



數位簽章：又有自我能夠產生，也可使別人驗證

驗證性 Authentication : Verify the msg from you

完整性 Integrity : No midification for a msg

不可否認性 Non-repudiation : cannot withdraw your sgn