class Solution:

def coinChange(self, coins: List[int], amount: int) -> int:

dp = collections.defaultdict(lambda: float("inf"))

dp[0] = 0

for A in range(1, amount+1):

dp[A] = min([1 + dp[A - coin] for coin in coins])

return -1 if math.isinf(dp[amount]) else dp[amount]

動態規劃 用小問題解決大問題

如Fibonacci

這題想法是

目標是1時需要多少硬幣，目標是2時需要多少硬幣，目標是3時需要多少硬幣….到目標硬幣數