Intro to DP. The Fibonacci sequence $\{f_i\}_{i=0,1,\ldots}$ is defined as:

$$f_i = \begin{cases} 1 & i = 0, 1 \\ f_{i-1} + f_{i-2} & i > 1 \end{cases}$$

Give an efficient algorithm `fib` taking in an integer input i such that $\texttt{fib}(i) = f_i$.

Tips Appreciated. You are a third year Purdue CS PhD grad student trying to buy a \$3.81 black coffee at Mochaland. Your entire life savings is in your left pocket in the form of a few pennies (1 cent), dimes (10 cent), and nickels (5 cent) coins.

One of your remaining joys in life is putting your hand in your pocket and fidgeting with the weight of the coins – you'd like to imagine that having more coins means that you are doing better financially. Hence, you would like to purchase this black coffee with as few coins as possible. More generally, assume you have an infinite number of each type of coin and suppose a coffee costs n cents. Give an efficient algorithm `change` taking in coffee cost n as input and outputs the minimum number of coins needed to reach cost *exactly* n.[1]

[1]Followups: (1) Counterintuitively, it may take less coins to buy a large cup of coffee instead of a small cup. If a large cup costs m cents, and a small cup costs n cents, determine if it takes less coins to reach the exact price of a small cup or a large cup; (2) Find the minimum number of coins needed to exactly buy a n cent cup of coffee when you have at most p pennies, k nickels, and d dimes for some integers $p, k, d \geq 0$ in $O(npkd)$ time. Can you do this in $O(3n)$ time?