Task 03

- 'minCoins' takes coins & amount as parameters
- initializes 'dp' array to keep track no of coins to make to target amount
- Best case → dp[0] = 0
- A loop is run to figure out min coins required & returned.

Task 02

- 'WaysCounted' takes 'n' as parameter
- Best case set to if $n \leq 1$; returns n
  else runs the code
- The code implements a variation of Fibonacci sequence.

Task 01

- imports hippap priority queue
- 'Union Find' implements disjoint set ds
- 'Find' performs 'Union Find' to finds the root of each elem
- 'Union' uses disjoined set data structure to attach tree with smaller rank to root of larger tree