Given any positive integer n, the goal is to find the least number of perfect squares that sum to it.

If n is 12, then the expected output should be 3 because there are 3 perfect squares that add to it, 4. 9 is less than 12 but 9+4 does not sum to 12, so the 3 4’s are valid.

I would need an int array the same size as the given integer, then begin to fill it with the maximum value

A nested for loop will be created to loop through the array, with the second one adding the value of I to the current j squared while both loops are less than or equal to the array length. Because I will be operating with the current j, j in the loop will initially be 1. The nested for loop will set the current i plus the current j squared to the smallest value and return the expected number of perfect squares

public int numSquares(int n) {

int[] dp = new int[n + 1];

Arrays.fill(dp, Integer.MAX\_VALUE);

dp[0] = 0;

for(int i = 0; i <= n; i++){

for(int j = 1; i + j \* j <= n; j++){

dp[i + j \* j] = Math.min(dp[i + j \* j], dp[i] + 1);

}

}

return dp[n];

}