LeetCode – Perfect Squares (Java)

Given a positive integer n, find the least number of perfect square numbers (for example, 1, 4, 9, 16, ...) which sum to n.

For example, given n = 12, return 3 because 12 = 4 + 4 + 4; given n = 13, return 2 because 13 = 4 + 9.

Java Solution

This is a dp problem. The key is to find the relation which is dp[i] = min(dp[i], dp[i-square]+1). For example, dp[5]=dp[4]+1=1+1=2.

```
public int numSquares(int n) {
    int max = (int) Math.sqrt(n);

    int[] dp = new int[n+1];
    Arrays.fill(dp, Integer.MAX_VALUE);

    for(int i=1; i<=n; i++){
        for(int j=1; j<=max; j++){
            if(i==j*j){
                dp[i]=1;
            }else if(i>j*j){
                dp[i]=Math.min(dp[i], dp[i-j*j] + 1);
            }
```

```
            }
        }

        return dp[n];
    }
```


You May Also Like ...

1. **LeetCode – Divide Two Integers (Java)**
2. **LeetCode – Triangle (Java)**
3. **LeetCode – Coin Change (Java)**
4. **LeetCode – Maximal Square (Java)**

Category >> Algorithms

If you want someone to read your code, please put the code inside <pre><code> and </code></pre> tags. For example:

```
<pre><code>
String foo = "bar";
</code></pre>
```