Find minimum number of coins

Problem Statement: Given a value V, if we want to make a change for V Rs, and we have an infinite supply of each of the denominations in Indian currency, i.e., we have an infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Examples:

Example 1:

Input: V = 70

Output: 2

Explaination: We need a 50 Rs note and a 20 Rs note.

Example 2:

Input: V = 121

Output: 3

Explaination: We need a 100 Rs note, a 20 Rs note and a 1 Rs coin.

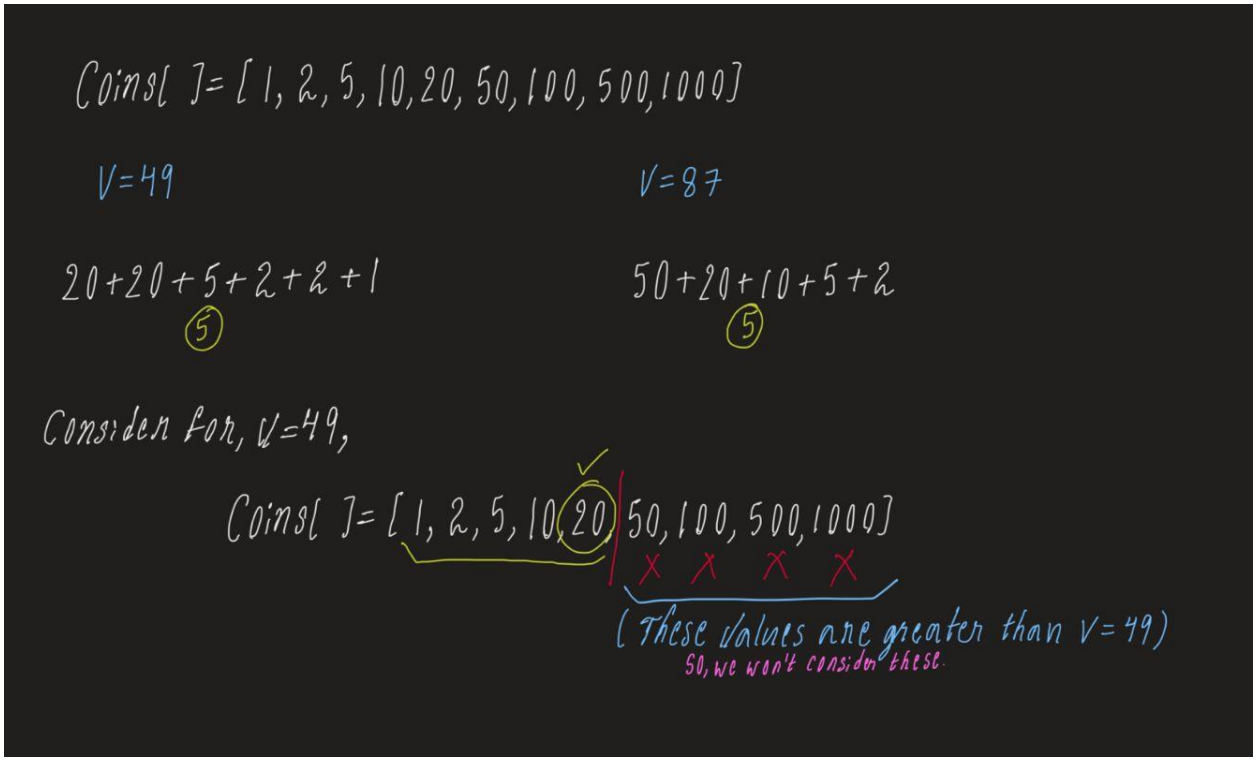
Solution:

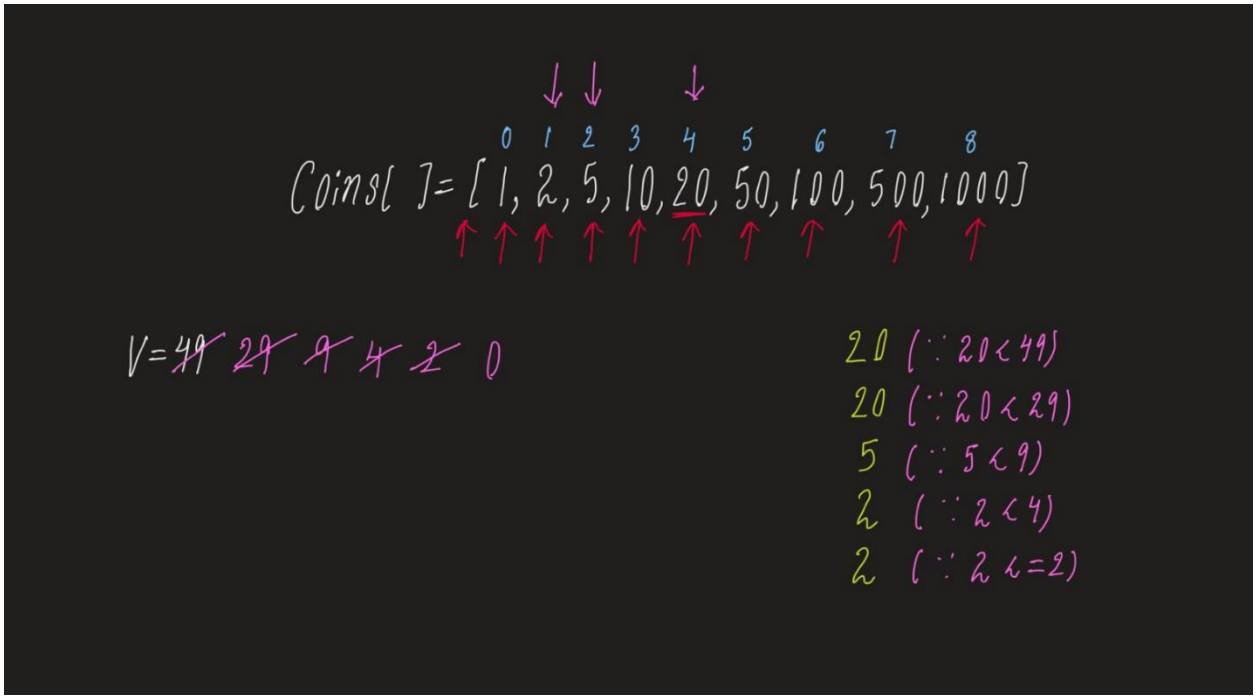
Disclaimer: *Don't jump directly to the solution, try it out yourself first.*

Solution: Greedy Algorithm

Approach: We will keep a pointer at the end of the array i. Now **while(V >= coins[i])** we will reduce V by coins[i] and add it to the ans array.

We will also ignore the coins which are greater than V and the coins which are less than V. We consider them and reduce the value of V by coins[I].





Code:

```
public static int findMinimumCoins(int amount)
    {
       int[] coins={1,2,5,10,20,50,100,500,1000};
       int totalCoins=0;
       int i=coins.length-1;
       while(amount>0&&i>=0)
       {
           if(amount>=coins[i])
           {
               totalCoins+=amount/coins[i];
               amount=amount%coins[i];
           }
           --i;
       }
    return totalCoins;
    }
```

Time Complexity:O(V)

Space Complexity:O(1)