Change-making problem

You have an unlimited number of pennies, dimes, and quarters. Given an integer $x \geq 0$, what's the minimum number of coins you can choose to sum to x exactly?

Recursive algorithm:

```
opt(i) {
    if i < 0:
        return ∞;
    if i = 0:
        return 0;
    return min(
                opt(i − 1)+1,
                opt(i − 10)+1,
                opt(i − 25)+1
                );
}
```

Dynamic Programming algorithm:

```
opt-DP(x) {
    d = [1,10,25];
    arr = new int[x + 1];
    arr[0] = 0;
    for i = 1 to x:
        arr[i] = ∞;
        for j = 1 to 3:
            if i − d_j ≥ 0  and  1 + arr[i − d_j] < arr[i]:
                arr[i] = 1 + arr[i − d_j];
    return arr[x];
}
```