算法设计与分析-work7

yu wang

2024 年 6 月 1 日

1 问题一

第一问，不可以贪心策略的算法解决问题。我们举一个很简单的例子，如果需要找零 10 元，根据贪心选择的性质，会首先找零 7 元一张，然后是三张一元的，最终合计为 4 张，而在实际找零我们只需要一张 6 元一张 4 元即可完成最少货币张数的找零，思考原因或许是几个货币类型不是倍数的原因。

下面将设计一个动态规划策略的算法解决问题：

其中 A 为一个货币数组，里面存放的是所有类型的货币，x 为需要找 x 元零钱

```
min-num(A,x)
1: Let f be a new array filled INT_MAX
2: f[0]=0
3: for i=1 to n do
4:     for j=1 to A.length()
5:         temp=A[j]
6:         if i>=temp and !(f[i-temp]==INT_MAX)
7:             f[i]=min(f[i],f[i-temp]+1)
8: return f[n]
```

证明正确性：

首先我们要证明该问题满足最优子解雇的性质，我们假设有一个最优的找零的方案，它能够求得找零的最小货币张数，则对于最后一张选择的纸币，我们有四种选择，1 元，4 元，6 元，7 元。

我们考虑最后一张纸币是一元的情况，我们可以得知，对于总数为 x-1 元的零钱，之前的选取纸币张数一定是最少的，否则我们一定可以找到一个更少的纸币张数的方案，使 X 元找零的张数是最少的，于是得到了这个找零方案是最优的矛盾，最终的找零张数为 x-1 元的找零张数加上 1。

接着我们考虑最后一张纸币是 4 元，6 元，7 元的情况，我们同样可以得到之前的找零方案一定是最优的。因此总而言之，该问题满足最优子结构的性质。

下面我们我们来证明算法的正确性:

首先，确定谓词 P（n）:

.P(n)：该算法能够求解出 n 元货币找零方案，使得所使用的货币总张数最少。

第二步是证明基本情况 P（0）和 P（1）：当找零为 0 元是，张数为 0。这是显然成立的，当只需要找零一元时，我们只需要找零一元一张。因此，成立。

第三步是证明一般情况 $\forall n \in N(P(0) \wedge P(1) \wedge \cdots \wedge P(n) \Rightarrow P(n+1))$，假设对于任意前 k ($0 \leq k \leq n$) 元，算法都能求解货币找零方案，使得所使用的货币总张数最少，记为 f[1...n] 。那么需要找零 n+1 时有四种情况，第一种最后一张找零为一元时，那么最后的找零张数为 f[n+1-1]+1, 第二种最后一张找零为 4 元时，那么最后的找零张数为 f[n+1-4]+1；第三种最后一张找零为 6 元时，那么最后的找零张数为 f[n+1-6]+1；第二种最后一张找零为 7 元时，那么最后的找零张数为 f[n+1-7]+1；比较这四种情况的最少零钱张数。于是我们求解出了零钱数为 n+1 时的最小找零张数。综上我们可以证明该算法能够求解出最优的找零方案，所使用的货币总张数最少。

2 问题二

、可以使用贪心策略的算法求解

下面将设计一个贪心策略的算法解决问题:

其中 A 为一个货币数组，里面存放的是所有类型的货币，x 为需要找 x 元零钱,sort 函数能够将数组从大到小降序排序

min-num(A,x)

1: sort(A)

2: count=0

```
for i=1 to A.length() do
    count+=n/A[i]
    n % =A[i]
return count
```

证明正确性：

首先我们要证明该问题满足最优子解雇的性质，我们假设有一个最优的找零的方案，它能够求得找零的最小货币张数，则对于最后一张选择的纸币，我们有四种选择，1 元，5 元，10 元，20 元。

我们考虑最后一张纸币是一元的情况，我们可以得知，对于总数为 x-1 元的零钱，之前的选取纸币张数一定是最少的，否则我们一定可以找到一个更少的纸币张数的方案，使 X 元找零的张数是最少的，于是得到了这个找零方案是最优的矛盾，最终的找零张数为 x-1 元的找零张数加上 1。

接着我们考虑最后一张纸币是 5 元，10 元，20 元的情况，我们同样可以得到之前的找零方案一定是最优的。因此总而言之，该问题满足最优子结构的性质。

接下来我们要证明贪心选择性的正确性。假设我们有一个找零张数最少的找零方案，其中选择出的货币面值最大不是理论可以选取的面值最大，即我们可以选择一个更大的面值货币且这个货币面值并没有超出零钱总数，我们将这个更大面值代替此面值的方案选择。由所有的货币类型可以，大的货币是小的货币的倍数，那么，对于这个更改的方案，它一定可以替换成一个张数更多选择货币面值更小的，由此我们便证明了此方案的更改，使得找零方案的货币张数减少了，因此我们的新的找零方案也是一个最优的找零方案，因此贪心选择性是正确的。

下面我们我们来证明算法的正确性:

首先，确定谓词 P（n）：

.P(n)：该算法能够求解出 n 元货币找零方案，使得所使用的货币总张数最少。

第二步是证明基本情况 P（0）和 P（1）：当找零为 0 元是，张数为 0。这是显然成立的，当只需要找零一元时，我们只需要找零一元一张。因此，成立。

第三步是证明一般情况 $\forall n \in N(P(0) \wedge P(1) \wedge \cdots \wedge P(n) \Rightarrow P(n+1))$，假设对于任意前 k ($0 \leq k \leq n$) 元，算法都能求解货币找零方案，使得所使用的货币总张数最少，记为 f[1...n] 。我们选取与前 n 个活动兼容的活动集

合 S' 由最优子结构的性质，总的最优找零方案一定包含在这个集合上的最优子方案, 此时如果 S' 选择的第一个活动如果不是面值可以选取的最大的那种货币，我们一定可以将其替换成可以选择的面值最大的货币，使得总的找零货币张数最小，所以 P(n+1) 成立，于是我们求解出了零钱数为 n+1 时的最小找零张数。综上我们可以证明该算法能够求解出最优的找零方案，所使用的货币总张数最少。