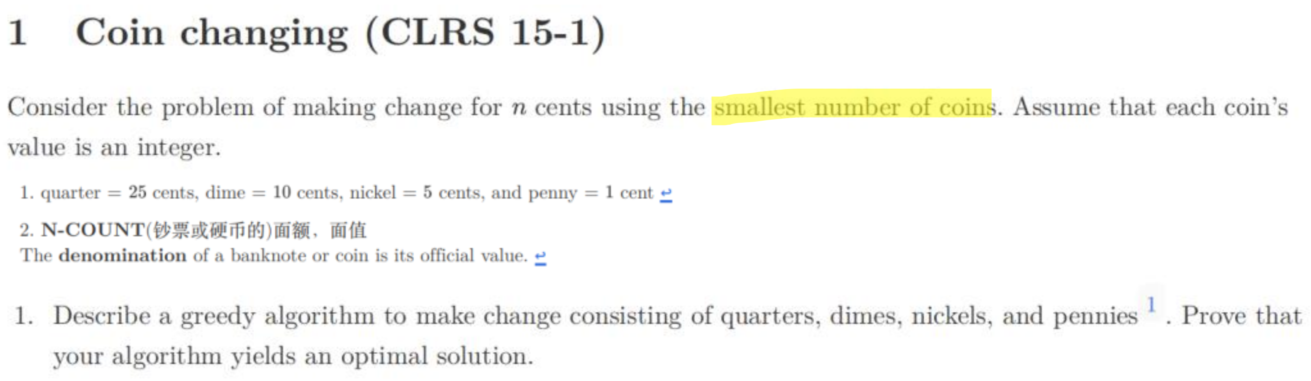
A8 毛九弢 221900175

2023年12月9日　　18:10



1 Coin changing (CLRS 15-1)

Consider the problem of making change for n cents using the smallest number of coins. Assume that each coin's value is an integer.

1. quarter = 25 cents, dime = 10 cents, nickel = 5 cents, and penny = 1 cent ↩
2. **N-COUNT**(钞票或硬币的)面额，面值
The **denomination** of a banknote or coin is its official value. ↩

1. Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies [1]. Prove that your algorithm yields an optimal solution.

贪心策略：

每一步选择不超过n的最大面值c的钞票，并把n-c，直到n=0；（显然算法会终止，因为存在面值为1的钞票）

正确性：

设问题答案也就是size = n 时候的最优解为$MinNumCoins(n)$

显然：$MinNumCoins(0) = 0$；$MinNumCoins(1) = 1$

证明：

1. 最优子结构：

   $MinNumCoins(n)$ = 1 + min($MinNumCoins(n - c)$ for c in [25,10,5,1] if $n \geq c$)

2. 存在贪心的选择：

   选择不超过n的最大面值c的钞票

证明：设最大不超过n的面值为c,

假设最优策略里没有c，那么对n进行讨论：

(1) $n < 5$: $c = 1$, 存在c

(2) $5 \leq n < 10$: 将5个penny替换为1个nickle变得更优，也存在c了

(3) $10 \leq n < 25$:

　一定存在某个p和n的组合使金额为10,将这个组合替换为一个dime更优了,也存在c了.

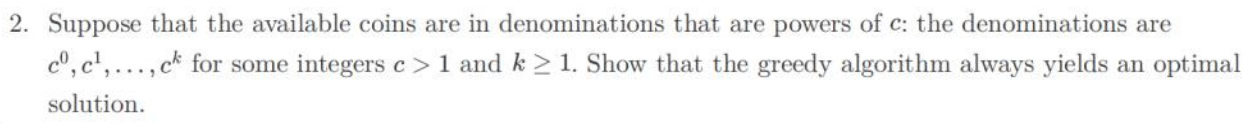
(4) $25 \leq n$:

　(Ⅰ) $nums\ of\ dimes \geq 3$: $3\ dimes \rightarrow 1\ quarter + 1\ nickel$

　(Ⅱ) $nums\ of\ dimes \leq 2$: 一定存在某个d、p和n的组合使金额为25,将这个组合替换为一个$quarter$更优了

于是也存在c了,

故而最优策略一定存在c,即存在贪心的选择



2. Suppose that the available coins are in denominations that are powers of c: the denominations are $c^0, c^1, \ldots, c^k$ for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.

证明：

1. 最优子结构：

   $MinNumCoins(n)$ = 1 + min($MinNumCoins(n - c^i)$ for i in range(k+1) if $n \geq c$)

2. 存在贪心的选择：

   选择不超过n的最大面值q的钞票

证明：设最大不超过n的面值为$q = c^i$,

　断言1：任意一个和为n的组合T，如果$c^m \leq n \leq c^{1+m} (m \leq k)$ and $\forall x \in T, x < c^m$

　　　则存在一个T的子组合Ts，s.t.$\sum_{x \text{ in } Ts'} x = c^m$

　证明断言1：考虑c进制这件事，使用c进制表示n是完备的且唯一的，且m位的c进制数小于c^m,

　　　故而，设T为按组合T中的数，表示的"c进制数"一定有部分位有溢出，

将这些溢出进位，也一定能进位到m+1位，得到T'，
那么显然也就存在了Ts。
故而当可以选择q时，如果存在一个不选择q的最优策略，
那么我们可以将这个最优策略的部分组合直接替换为一个q，得到更优的策略。
所以不存在一个不选择q的最优策略。
于是贪心就是最优解。

3. Give a set of coin denominations [2] for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every value of n.

问题二中，我们思考到了进制的话题，
那么我们假设这些coins的值由小到大为$c_0, c_1 \dots c_k$,
设序列$x_0, x_1 \dots x_k$为$coins$的个数，
另设进位系数：$a_0, a_1 \dots a_k$, 其中$a_i = \frac{c_{i+1}}{c_i}$ $if\ i < k, else\ inf$.
定义进位如果$x_i \geq a_i$ 则让x_i $-=$ a_i; x_{i+1} $+= 1$ 后根据差值由x_{i-1}向x_0调整.

贪心的本质就在于进位导致最优
也就是当进位发生后$\sum_i x_i$会变小那么我们只要构造一个进位导致和变大的coins序列即可
很容易我们得到：1，3，4；当n = 6时，贪心失败。

4. Give an $O(nk)$-time algorithm that makes change for any set of k different coin denominations using the smallest number of coins, assuming that one of the coins is a penny.

```python
from typing import List
inf = 0x7f7f7f7f
def changeWithSmallestCoins(coin_values:List[int], change:int)->int:
    dp = [inf for _ in range(change + 1)] # dp[i] 表示 找i钱的个数
    dp[0] = 0
    for i in range(1,change + 1):
        for c in coin_values:
            if c > i or dp[i-c] >= inf: continue
            dp[i] = min(dp[i-c]+1,dp[i])
    return dp[change] if dp[change] < inf else -1
if __name__ == "__main__":
    coins = list(map(int,input().split(sep=" ")))
    change = int(input())
    print(changeWithSmallestCoins(coins,change))
```

2 Yen's improvement to Bellman-Ford (CLRS 22-1)

The Bellman-Ford algorithm does not specify the order in which to relax edges in each pass. Consider the following method for deciding upon the order. Before the first pass, assign an arbitrary linear order $v_1, v_2, \dots, v_{|V|}$ to the vertices of the input graph $G = (V, E)$. Then partition the edge set E into $E_f \cup E_b$, where $E_f = \{(v_i, v_j) \in E : i < j\}$ and $E_b = \{(v_i, v_j) \in E : i > j\}$ (Assume that G contains no self-loops, so that every edge belongs to either E_f or E_b. Define $G_f = (V, E_f)$ and $G_b = (V, E_b)$.

1. Prove that G_f is acyclic with topological sort $\langle v_1, v_2, \ldots, v_{|V|}\rangle$ and that G_b is acyclic with topological sort $\langle v_{|V|}, v_{|V|-1}, \ldots, v_1\rangle$.

Suppose that each pass of the Bellman-Ford algorithm relaxes edges in the following way. First, visit each vertex in the order $v_1, v_2, \ldots, v_{|V|}$, relaxing edges of that leave the vertex. Then visit each vertex in the order $v_{|V|}, v_{|V|-1}, \ldots, v_1$, relaxing edges of that leave the vertex.

由于拓扑序意味，vi在vj前 当且仅当 vj没有向vi的通路，如果上述不是拓扑序，则
在G_f中存在$i < j$，且vj有向vi的通路，则一定存在$v_x, v_y, s.t. x > y, \left(v_x, v_y\right) \in E_f$ *矛盾！*
$\mathrm{G_b}$亦然。

2. Prove that with this scheme, if G contains no negative-weight cycles that are reachable from the source vertex s, then after only $\lceil |V|/2 \rceil$ passes over the edges, $v.d = \delta(s, v)$ for all vertices $v \in V$.

不妨设图连通，否则 $\mathrm{v.d = inf}$ ⬚ $\mathrm{iff.}\ \delta(\mathrm{s,v}) = \ \mathrm{inf}$
任意取一条从s到v的最短路径为$\mathrm{v_{i_0}, v_{i_1} \ldots v_{i_m}}$，我们需要再次搜点当且仅当$\left(\mathrm{i_j < i_{j+1}}\right)$ ^ $\left(\mathrm{i_{j+1} < i_{j+2}}\right) = 1$(⬚ ^ 为异或)
而这样的次数最多为$\frac{(m+1+1)}{2}$ 又由于$\mathrm{m < |V|}$故而：只要对全部边搜$\left\lceil\frac{|V|}{2}\right\rceil$ *的次数即可*

3. Does this scheme improve the asymptotic running time of the Bellman-Ford algorithm?

它并没有改进Bellman-ford算法的asymptotic运行时间，
它只是将系数降低到1/2。时间复杂度仍为O(|E|*|V|)。