5. 找零问题：假设有 m 种面值不同的硬币，每个面值存于数组 A[1..m]中，现在用这些硬币来找钱，各种硬币的使用个数不限。对于给定的金额 N，最少可以由几枚硬币组成？所需的各种面值的硬币分别是多少？

（1）对于找零问题，请给出一个实例，使得贪心算法不能输出一个最优解。所给的实例应当包括面值为 1 的硬币，以保证对任意的 N 值都是有解的；（4 分）

（2）为找零问题写一个贪心算法的伪代码,它以金额 N 和硬币的面值 d1>d2>…>dm 作为输入，以每种面值硬币的数量构成的数组 D[1..m]作为输出或输出无解；（8 分）

（3）该算法的时间效率是多少？（3 分）

解答：

（1）给定面值 10，5，4，1，和金额 18，利用贪心算法得到的解为 D[1,1,0,3]，一共 5 枚硬币，但实际的最优解为 D[1,0,2,0]，一共 3 枚硬币。

（2）伪码描述如下：

```
Algorithms Change(n,D[1..m])
//用贪心算法求找零问题
//输入：非负整数 n,硬币面值存放在数组 D
//输出：数组 A[1..m]，其中每个单元存放每种面值硬币的数量,或者输出无解
对数组 D 按降序进行排序；
for i←1 to m do
   A[i] ←⌊n/D[i]⌋;
   n ←n mod D[i];
end for
if n==0 return A；
else print(“no solution”);
```

（3）算法的时间效率取决于排序的时间效率，故为 Θ(mlogm).,

6. 多机调度问题要求给出一种作业调度方案，使给定的 n 个作业在尽可能短的时间内由 m 台机器加工处理完成。其中约定，每个作业均可在任何一台机器上加工处理，但未完工前不允许中断处理。作业不能拆分成更小的子作业。

（1） 请编写一个贪心算法找出这个调度方案，使得 n 个作业的总耗费时间尽可能少。（8 分）

（2） 请分析该算法在 $m \geq n$ 和 $m < n$ 的情况下针对问题规模 n 的时间复杂度。（4 分）

（3） 针对以下多机调度问题的实例，你的算法得到的调度方案是怎样的？作业平均周转时间是多少？ （4 分）

m= 3, n= 7

编号	1	2	3	4	5	6	7
时间	2	5	14	17	3	11	9

解答：

（1）采用最长处理时间作业优先的贪心策略:

当 n≤m 时, 只要将机器 i 的[0, ti]时间区间分配给作业 i 即可。

当 n>m 时, 将 n 个作业依其所需的处理时间从大到小排序,然后依次将作业分配给空闲的处理机。

```
Algorithms Schedule(m, n, D[1..n])
//用贪心算法求多机调度问题
//输入：非负整数 m 和 n，作业运行时间存放在数组 D 中，m 为机器数，n 为作业数
//输出：每个机器上分配的作业编号
  if(m>=n)
     为每个作业分配一台机器；
     return;
  end if
  Sort(D,n);      //对 n 个作业时间按降序排序
```

将每台机器的运行时间初始化为 0；
以机器的运行时间为键值，初始化大小为 m 的小顶堆；
for i←n to 1 do
  从堆中删除运行时间 Tj 最小的机器 j；
  Tj ←Tj+D[i]；
  输出“将机器 j 指派给作业‘D[i]的编号’”
  以运行时间 Tj 为健值将机器 j 插入小顶堆；
end for

（2）若 m>=n，则 n 个作业每个作业均可分配一台机器，贪心算法的时间效率为Θ(1)；

若 m<n，则贪心算法的主要时间取决于对作业排序花费的时间 O(nlogn)，建初始堆花费 O(m)，插入堆花费.O(nlogm)，因此最终的时间效率为 O(nlogn)。

（3） 调度方案为：

机器 1：作业 4、作业 5

机器 2：作业 3、作业 2、作业 1

机器 3：作业 6、作业 7

作业总的时间耗费为：21，平均周转时间为：3