1. 水仙花数问题

水仙花数是指一个 n 位数（n≥3），它的每个位上的数字的 n 次幂之和等于它本身。本文将通过Python代码实现打印水仙花数，具体如下：

```
#水仙花数
#narcissistic number
#水仙花数是指一个 n 位数 ( n≥3 )，它的每个位上的数字的 n 次幂之和等于它本身。
#（例如：1^3 + 5^3+ 3^3 = 153)
import math
import string

for x in range(1,10):
  a=x*x*x
  for y in range(0,10):
    b=y*y*y
    for z in range(0,10):
      c=z*z*z
      d=a+b+c
      w='%d' %x+'%d' %y+'%d' %z
      if d==int(w):
        print('水仙花数：'+w+'\n')
```

方法二：

```
for x in range(100, 1000):
        # 取个位数
        a = x % 10
        # 取十位数
        b = int(x % 100 / 10)
        # 取百位数
        c = int(x / 100)
        if a**3 + b**3 + c**3 == x:
            print(x)
```

2. 快速排序问题

```
def quicksort(array):
  if len(array) < 2:
    # base case, arrays with 0 or 1 element are already "sorted"
    return array
  else:
    # recursive case
    pivot = array[0]
    # sub-array of all the elements less than the pivot
    less = [i for i in array[1:] if i <= pivot]
    # sub-array of all the elements greater than the pivot
    greater = [i for i in array[1:] if i > pivot]
    return quicksort(less) + [pivot] + quicksort(greater)

print(quicksort([100, 50, 20, 3]))
```

在这个问题中我们用到了**递归**的方法

3.找水果商问题（最短路径问题)

```
from collections import deque

def person_is_seller(name):
     return name[-1] == 'm'
# 用字典（散队列来创建关系图）
graph = {}
graph["you"] = ["alice", "bob", "claire"]
graph["bob"] = ["anuj", "peggy"]
graph["alice"] = ["peggy"]
graph["claire"] = ["thom", "jonny"]
graph["anuj"] = []
graph["peggy"] = []
graph["thom"] = []
graph["jonny"] = []

def search(name):
   search_queue = deque()
   search_queue += graph[name]
   # This array is how you keep track of which people you've searched before.
   searched = []
   while search_queue:
       person = search_queue.popleft()
       # Only search this person if you haven't already searched them.
       if person not in searched:
           if person_is_seller(person):
               print(person + " is a mango seller!")
               return True
           else:
               search_queue += graph[person]
               # Marks this person as searched
               searched.append(person)
   return False

search("you")
```