Mathematics for Computer Science: Homework 8

Xingyu Su 2015010697

April 18, 2019

Special Problem 1

In class we discussed the *Mr. P and Mr. S problem*, in which a conversation with 4 rounds of communications takes place. Let $2 \leq m < 98n \leq 99$. Of the $\binom{98}{2} = 4753$ pairs of (m, n) in the range, let a_j be the number of pairs leading to this conversation (exactly as stated in class) stopping (and unable to continue) after exactly j rounds for $j = 0, 1, 2, 3, 4$, respectively. You should get $a_4 = 1$.

Mr. P and Mr. S problem: *Two numbers m and n are chosen such that $2 \leq m \leq n \leq 99$. Mr. S is told their sum and Mr. P is told their product. The following dialogue ensues:*

Mr. P: I don't know the numbers.

Mr. S: I knew you didn't know. I don't know either.

Mr. P: Now I know the numbers.

Mr. S: Now I know them too.

Question: Write a computer program and determine the values of a_0, a_1, a_2, a_3, a_4. You should give a concise explanation of the principles of your design.

Answer:

It's not difficult to analyze this problem. Step by step we can get all a_j.

(1) To get a_0, we need to know exactly when will *Mr. P don't know the numbers.* That means, the product given to him is not unique of products in all pertubations. Define the whole produts array as *muls*. Count them one by one, we will know the times each *mul* shows up. Then select all situations that *mul* more than one time, we get a_0 situations left.

(2) To get a_1, we need to know when *Mr. S knew Mr. P didn't know.* Define the whole sums array as *sums*. All situations that *mul* shows exactly one time will make *Mr. P know the numbers.* So *Mr. S* knows that the *sum* given to him will not lead to situation like that. Which means, his *sum* does not belong to the *sums* that corresponding *muls* are unique. Here we get a_1.

(3) To get a_2, we need to know when *Mr. S still don't know the numbers.* First we should reduce the scope of results in to a_1. Then similar to (1), counting all *sums* and the ones not unique are which the numbers could belong to. a_2 is the number of these situations.

(4) a_3 is the number of situations that *Mr. P know the results* under all situations represented by a_2. So the product *Mr. P* got is unique in a_2 situations. Counting this, we get a_3 now.

(5) *Mr. S now knows the results* lead to similar counting process in (4). Counting all *sums* in a_3 and find all unique ones. We finally get a_4.

The program is clear now. I choose *python* as programming language since it's easy to do counting and selection. To get counting part seperated, I used *collections.Counter* function and mapping to write a *IndexFilter* that can get all index that corresponding values in $fvals$ satisifying *condition* of counting *cvals*.

The code I wrote is as below:

```python
#/usr/bin/env python
import numpy as np
from collections import Counter

def IndexFilter(cvals, fvals, condition=lambda fvi: True):
    counter = Counter(cvals)
    return [i for i,fvi in enumerate(fvals) if condition(counter[fvi])]

if __name__ == '__main__':
    # initialize
    numbers = np.array([(m,n) for m in range(2,100) for n in range(m+1,100)])
    sums = np.array([m+n for (m,n) in numbers])
    muls = np.array([m*n for (m,n) in numbers])

    # Subsets: when having no infomation, Mr.P don't know
    idx_a0 = IndexFilter(muls, muls, condition=lambda fmi: fmi>1)
    idx_a0_op = IndexFilter(muls, muls, condition=lambda fmi: fmi==1)

    # Subsets: Mr.S know Mr.P don't know
    opps = sums[idx_a0_op]
    idx_a1 = IndexFilter(opps, sums, condition=lambda fsi: fsi==0)

    # Subsets: Mr.S still don't know.
    sums,muls = sums[idx_a1],muls[idx_a1]
    idx_a2 = IndexFilter(sums, sums, condition=lambda fsi: fsi>1)

    # Subsets: Mr.P now know the answer.
    sums,muls = sums[idx_a2],muls[idx_a2]
    idx_a3 = IndexFilter(muls, muls, condition=lambda fmi: fmi==1)

    # Subsets: Mr.S now know the answer.
    sums,muls = sums[idx_a3],muls[idx_a3]
    idx_a4 = IndexFilter(sums, sums, condition=lambda fsi: fsi==1)

    # show results
    m,n = numbers[idx_a1][idx_a2][idx_a3][idx_a4][0]
    print("a0 = %d"%len(idx_a0))
    print("a1 = %d"%len(idx_a1))
    print("a2 = %d"%len(idx_a2))
    print("a3 = %d"%len(idx_a3))
    print("a4 = %d"%len(idx_a4))
    print("m = %d, n = %d"%(m,n))
```

The results of a_j are:

$$a_0 = 3021,\ a_1 = 145,\ a_2 = 145,\ a_3 = 86,\ a_4 = 1.$$

and the numbers are:

$$m = 4, n = 13, m + n = 17, m \times n = 52.$$

Special Problem 2

Let $\mathcal{C}$ be the unit circle on the complex plane, traversed counter-clockwise. Let n be any integer (positive, negative and zero), and $A_n = \oint_{\mathcal{C}} z^n$. $\mathcal{C}$ Determine the value of A_n. You should give a justification of your answer.

Answer:

To solve the integral along with unit circle on the complex plane, define the unit circle as $\gamma(\theta) = e^{i\theta}$. So:

$$\begin{aligned} A_n = \oint_{\mathcal{C}} z^n &= \int_0^{2\pi} e^{in\theta} de^{i\theta} \\ &= \int_0^{2\pi} ie^{(n+1)\theta} d\theta \\ &= i\int_0^{2\pi} \cos[(n+1)\theta]d\theta - \int_0^{2\pi} \sin[(n+1)\theta]d\theta \end{aligned}$$

Easy to know that when $n+1 \neq 0$, $\int_0^{2\pi} \cos[(n+1)\theta]d\theta = \int_0^{2\pi} \sin[(n+1)\theta]d\theta = 0$. So $A_n = 0$, for $n \neq -1$. When $n = -1$, we have:

$$A_n = \int_0^{2\pi} id\theta = 2\pi i$$

So, the results are:

$$A_n = \begin{cases} 2\pi i, & n = -1; \\ 0, & else. \end{cases}$$

For justification, consider the Cauchy integral theorem that:

When f is a holomorphic function, and let $\mathcal{C}$ be a rectifiable path whose start point is equal to its end point. Then

$$\oint_{\mathcal{C}} f(z)dz = 0.$$

For all $n \geq 0$, $f(z) = z^n$ is obvious holomorphic, so $A_n = 0$ for $n = 0, 1, 2, \cdots$. The method used before for solving $n = -1$ is totally calculus and is unnecessary to be discussed here. For the cases that $n < -1$. We know the Residue theorem that:

$$\oint_{\mathcal{C}} f(z)dz = 2\pi i \sum_{k=1}^{n} Res(f, z_k)$$

where $\mathcal{C}$ is a positively oriented simple closed curve and z_k are poles. And the residue:

$$Res(f, z_k) = \lim_{z \to z_k} \frac{1}{(k-1)!} \frac{d^{k-1}}{dz^{k-1}} (z-c)^k f(z)$$

So for $f(z) = z^{-n}, n > 1$, we have an n order pole $z = 0$.

$$A_n = \oint_{\mathcal{C}} f(z)dz = 2\pi i \lim_{z \to 0} \frac{1}{(n-1)!} \frac{d^{n-1}}{dz^{n-1}} z^n z^{-n} = 0.$$

In summary, the results still are:

$$A_n = \begin{cases} 2\pi i, & n = -1; \\ 0, & else. \end{cases}$$

Acknowledgement:

Thanks to Cauchy integral theorem and Residue theorem for SP2.