Discrete Mathematics: Homework 8

Name ID: Number

2020.4.29

1. Let $Z^2 = (a,b) : a,b \in \mathbb{Z}$. Define $\bigoplus$ and $\bigotimes$ over $\mathbb{Z}^2$ such that,

   (a) $(a,b)\bigoplus(c,d) = (a+c, b+d)$; and

   (b) $(a,b)\bigotimes(c,d) = (ac, bd)$.

   for all $(a,b),(c,d) \in \mathbb{Z}^2$. Prove or disprove that $(\mathbb{Z}^2, \bigoplus, \bigotimes)$ is a ring.

   证明.

   (a) Proof $(\mathbb{Z}^2, \bigoplus)$ is an Abelian group.

      i. Closure: $\forall (a,b),(c,d) \in \mathbb{Z}^2, (a,b)\bigoplus(c,d) = (a+c, b+d) \in \mathbb{Z}^2$

      ii. Asssociative: $\forall (a,b),(c,d),(e,f) \in \mathbb{Z}^2$,
      $((a,b)\bigoplus(c,d))\bigoplus(e,f) = (a+c, b+d)\bigoplus(e,f) = (a+c+e, b+d+f) = (a,b)\bigoplus(c+e, d+f) = (a,b)\bigoplus((c,d)\bigoplus(e,f))$

      iii. Identity: $\exists (0,0) \in \mathbb{Z}^2, \forall (a,b) \in \mathbb{Z}^2, (a,b)\bigoplus(0,0) = (a,b) = (0,0)\bigoplus(a,b)$

      iv. Inverse: $\forall (a,b) \in \mathbb{Z}^2, \exists (-a,-b) \in \mathbb{Z}^2, (a,b)\bigoplus(-a,-b) = (0,0)$

   (b) Proof $(\mathbb{Z}^2, \bigotimes)$ satisfies the property of closure and asssociativity

      i. Closure: $\forall (a,b),(c,d) \in \mathbb{Z}^2$,$(a,b)\bigotimes(c,d) = (ac, bd) \in \mathbb{Z}^2$

      ii. Asssociativity: $\forall (a,b),(c,d),(e,f) \in \mathbb{Z}^2$,
      $((a,b)\bigotimes(c,d))\bigotimes(e,f) = (ac,bd)\bigotimes(e,f) = (ace, bdf) = (a,b)\bigotimes(ce,df) = (a,b)\bigotimes((c,d)\bigotimes(e,f))$

   (c) Distributive Law:

      i. $\forall (a,b),(c,d),(e,f) \in \mathbb{Z}^2$,
      $(a,b)\bigotimes((c,d)\bigoplus(e,f)) = (a,b)\bigotimes(c+e, d+f) = (ac+ae, bd+bf)$
      $= (ac,bd)\bigoplus(ae,bf) = ((a,b)\bigotimes(c,d))\bigoplus((a,b)\bigotimes(e,f))$

      ii. $\forall (a,b),(c,d),(e,f) \in \mathbb{Z}^2$,
      we have $((a,b)\bigoplus(c,d))\bigotimes(e,f) = ((a,b)\bigotimes(e,f))\bigoplus((c,d)\bigotimes(e,f))$
      The proof is similiar

□

2. Let $\mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$ be the set of allpolynomials over $\mathbb{Z}$. Prove or disprove that $(\mathbb{Z}[X], +, \cdot)$ is a ring, where $+$ and $\cdot$ are the additionand the multiplication of polynomials over $\mathbb{Z}$.

证明.

(a) Prove $(\mathbb{Z}[X], +)$ is an Abelian group.

i. Closure:

Let $A \in \mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$

Let $B \in \mathbb{Z}[X] = \{g_0 + g_1X + \cdots + g_eX^e : e \geq 0, g_0, g_1, \ldots, g_e \in \mathbb{Z}\}$

W.L.O.G, Suppose $e > f$

$A + B = \{(f_0 + g_0) + (f_1 + g_1)X + \cdots + (f_d + g_d)X^d + \cdots + g_eX^e\} \in \mathbb{Z}[X]$

ii. Asssociative:

W.L.O.G, Suppose $f < g < e$

Let $A \in \mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$

Let $B \in \mathbb{Z}[X] = \{g_0 + g_1X + \cdots + g_eX^e : e \geq 0, g_0, g_1, \ldots, g_e \in \mathbb{Z}\}$

Let $C \in \mathbb{Z}[X] = \{h_0 + h_1X + \cdots + h_iX^i : i \geq 0, h_0, h_1, \ldots, h_i \in \mathbb{Z}\}$

$(A + B) + C = (f_0 + g_0) + (f_1 + g_1)X + \cdots + (f_d + g_d)X^d + \cdots + g_eX^e + h_0 + h_1X + \cdots + h_iX^i = (f_0 + g_0 + h_0) + (f_1 + g_1 + h_1)X + \cdots + (f_d + g_d + h_d)X^d + \ldots (g_e + h_e)X^e + \ldots h_iX^i$

$A + (B + C) = f_0 + f_1X + \cdots + f_dX^d + ((g_0 + h_0) + (g_1 + h_1)X + \cdots + (g_e + h_e)X^e + \ldots h_iX^i) = (f_0 + g_0 + h_0) + (f_1 + g_1 + h_1)X + \cdots + (f_d + g_d + h_d)X^d + \ldots (g_e + h_e)X^e + \ldots h_iX^i$

iii. Identity:

$\exists I = \{0 + 0 + \cdots + 0\} \in \mathbb{Z}[X], \forall A \in \mathbb{Z}[X], A + I = A$

iv. Inverse: $\forall A \in \mathbb{Z}[X] \exists - A \in \mathbb{Z}[X]$, such that $A + (-A) = I$

(b) Proof $(\mathbb{Z}[X], \cdot)$ satisfies the property of closure and asssociativity

i. Closure

Let $A \in \mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$

Let $B \in \mathbb{Z}[X] = \{g_0 + g_1X + \cdots + g_eX^e : e \geq 0, g_0, g_1, \ldots, g_e \in \mathbb{Z}\}$

W.L.O.G, Suppose $e > f$

$A \cdot B = (f_0 \cdot g_0) + (f_1g_0 + f_0g_1)X \ldots (f_dg_e)X^{d+e} \in \mathbb{Z}[X]$

ii. Asssociativity W.L.O.G, Suppose $f < g < e$

Let $A \in \mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$

Let $B \in \mathbb{Z}[X] = \{g_0 + g_1X + \cdots + g_eX^e : e \geq 0, g_0, g_1, \ldots, g_e \in \mathbb{Z}\}$

Let $C \in \mathbb{Z}[X] = \{h_0 + h_1X + \cdots + h_iX^i : i \geq 0, h_0, h_1, \ldots, h_i \in \mathbb{Z}\}$

$(A \cdot B) \cdot C = (f_0 \cdot g_0) + (f_1g_0 + f_0g_1)X \ldots (f_dg_e)X^{d+e} \cdot C = (f_0 \cdot g_0 \cdot h_0) + \cdots + (f_d \cdot g_e \cdot h_i)X^{d+e+i}$

Similiarly, we can prove $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

(c) Distributive Law: W.L.O.G, Suppose $f < g < e$

Let $A \in \mathbb{Z}[X] = \{f_0 + f_1X + \cdots + f_dX^d : d \geq 0, f_0, f_1, \ldots, f_d \in \mathbb{Z}\}$

Let $B \in \mathbb{Z}[X] = \{g_0 + g_1X + \cdots + g_eX^e : e \geq 0, g_0, g_1, \ldots, g_e \in \mathbb{Z}\}$

Let $C \in \mathbb{Z}[X] = \{h_0 + h_1X + \cdots + h_iX^i : i \geq 0, h_0, h_1, \ldots, h_i \in \mathbb{Z}\}$

$A \cdot (B + C) = A \cdot (\sum_{m=0}^{e} X^m + \sum_{n=e+1}^{i} X^n) = f_0g_0h_0 + \cdots + f_dh_iX^{d+f}$

$ab + ac =_0 g_0h_0 + \cdots + f_dh_iX^{d+f}$

Similiarly, $(A + B) \cdot C = ac + bc$

□

3. Write a computer program to reconstruct the secret $s \in \mathbb{Z}_{1125899906900597}$ in Shamir's $(5, 9) - \textit{threshold}$ secret sharing scheme. In the devised program, the reconstruction of s shouldbe based on the Lagrange interpolation formula. Use your program to reconstruct the secret s, given that the 9 shares are as follows:

i	s_i
1	75044643784737
2	940519894412855
3	941263003333598
4	736739711411826
5	254180887785524
6	940382343666996
7	132205297839880
8	63775631863924
9	1111084448671404

Solution:

The Code is as follows,

```
import collections
import sys
sys.setrecursionlimit(10000000) #Enlarge maximum recursion depth

def Exgcd(a: int, b: int):
    '''
          gcd(a,b)=ax+by
    '''
    if (not b):
```

```python
        return [1, 0]
    x, y = Exgcd(b, a % b)
    return [y, x - a // b * y]

def inv(x: int, mod: int):
        return Exgcd(x, mod)[0] % mod

def TSSS(mod: int, **args) -> int:
        '''
        args: dic{}

        return secret s = f(0) by Lagrange Interpolation
        '''
        s = 0
        for i in args:
                terms = 1 #Terms of each f(i)delta_i(0)
                deno = 1 #Denominator
                for j in args:
                        if i == j:
                                continue
                        terms = terms * (0 - int(j))
                        deno = deno * (int(i) -int(j))
                s += args[i] * terms * inv(deno, mod)
                s %= mod
        return s

d = {
    '1': 75044643784737,
    '2': 940519894412855,
    '3': 941263003333598,
    '4': 736739711411826,
    '5': 254180887785524
}

print(TSSS(1125899906900597, **d))
```

And the Output is :

330836359559300

4. An officer stored in his safe a very important letter. He shared the password $s \in \mathbb{Z}_{1125899906900597}$ to the safe among 9 soldiers using $Shamir's(5,9) - threshold$ secret shar-ing scheme. After the officer was killed in a battle, the 9 soldiers need to open the safe. Supposethat they provided the following shares in the reconstruction process:

i	s_i
1	150550125355646
2	944474507418938
3	110040335185999
4	676042268761809
5	193274108888331
6	904128547609081
7	354197665334455
8	416432161112962
9	283942097426448

Among the 9 soldiers 2 were spies and provided wrong shares in order to prevent the othersoldiers from opening the safe. Use your computer program in Question 3 to find the spies andthen recover the password s from the correct shares。

Solution:

The Code is as follows:

```python
import collections
import sys
from itertools import combinations, permutations
sys.setrecursionlimit(10000000) #Enlarge maximum recursion depth

def Exgcd(a: int, b: int):
    '''
          gcd(a,b)=ax+by
    '''
    if (not b):
        return [1, 0]
    x, y = Exgcd(b, a % b)
    return [y, x - a // b * y]

def inv(x: int, mod: int):
        return Exgcd(x, mod)[0] % mod

def TSSS(mod: int, **args) -> int:
        '''
```

```
        args: dic{}

        return secret s = f(0) by Lagrange Interpolation
        '''
        s = 0
        for i in args:
                terms = 1 #Terms of each f(i)delta_i(0)
                deno = 1 #Denominator
                for j in args:
                        if i == j:
                                continue
                        terms = terms * (0 - int(j))
                        deno = deno * (int(i) -int(j))
                s += args[i] * terms * inv(deno, mod)
                s %= mod
        return s

def remove(list:list, element):
        for i in list:
                if i == element:
                        list.remove(i)



d = {
    '1': 150550125355646,
    '2': 944474507418938,
    '3': 110040335185999,
    '4': 676042268761809,
    '5': 193274108888331,
    '6': 904128547609081,
    '7': 354197665334455,
    '8': 416432161112962,
    '9': 283942097426448
}



C = list(combinations([1, 2, 3, 4, 5, 6, 7, 8, 9], 5))
num = [1,2,3,4,5,6,7,8,9]
list = list()
for i in C:
        dic = dict()
        for j in range(5):
                dic[str(i[j])] = d[str(i[j])]
        print(i)
```

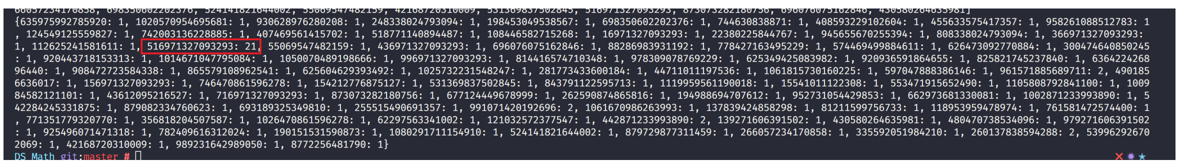
```
            print(TSSS(1125899906900597, **dic))
            list.append(TSSS(1125899906900597, **dic))
            print()
            if (TSSS(1125899906900597, **dic) == 516971327093293):
                for j in range(5):
                    remove(num,i[j])
print(num)

# print(list)
# dict_num = {}
# for item in list:
#     if item not in dict_num.keys():
#         dict_num[item] = list.count(item)

# import operator
# sorted(dict_num.items(),key=operator.itemgetter(1))

# print (dict_num)
```

The Output:



```
{635975992785920: 1, 1020570954695681: 1, 930628976280208: 1, 248338024793094: 1, 198453049538567: 1, 698350602202376: 1, 744630838871: 1, 408593229102604: 1, 455633575417357: 1, 958261088512783: 1
, 124549125559827: 1, 742003136228885: 1, 407469561415702: 1, 518771140894487: 1, 108446582715268: 1, 16971327093293: 1, 22380225844767: 1, 945655670255394: 1, 808338024793094: 1, 366971327093293:
1, 112625241581611: 1, 516971327093293: 21, 55069547482159: 1, 436971327093293: 1, 696076075162846: 1, 88286983931192: 1, 778427163495229: 1, 574469499884611: 1, 626473092770884: 1, 300474640850245
: 1, 920443718153313: 1, 1014671047795084: 1, 1050070489198666: 1, 996971327093293: 1, 814416574710348: 1, 978309078769229: 1, 625349425083982: 1, 920936591864655: 1, 825821745237840: 1, 6364224268
96440: 1, 908472723584338: 1, 865579108962541: 1, 625604629393492: 1, 1025732231548247: 1, 281773433600184: 1, 44711011197536: 1, 1061815730160225: 1, 597047888386146: 1, 961571885689711: 2, 490185
6636017: 1, 156971327093293: 1, 746470861596278: 1, 154212776875127: 1, 531369837502845: 1, 843791122595713: 1, 1119959561190018: 1, 15541011122308: 1, 553471915652490: 1, 1105808792841100: 1, 1009
84582121101: 1, 43612095216527: 1, 716971327093293: 1, 873073282180756: 1, 677124449678999: 1, 262590874865816: 1, 194988694707612: 1, 952731054429853: 1, 662973681330081: 1, 1002871233993890: 1, 5
42284245331875: 1, 879082334760623: 1, 693189325349810: 1, 255515490691357: 1, 991071420192696: 2, 1061670986263993: 1, 137839424858298: 1, 81211599756733: 1, 118953959478974: 1, 761581472574400: 1
, 771351779320770: 1, 356818204507587: 1, 1026470861596278: 1, 62297563341002: 1, 121032572377547: 1, 442871233993890: 2, 139271606391502: 1, 430580264635981: 1, 480470738534096: 1, 979271606391502
: 1, 925496071471318: 1, 782409616312024: 1, 190151531590873: 1, 1080291711154910: 1, 524141821644002: 1, 879729877311459: 1, 266057234170858: 1, 335592051984210: 1, 260137838594288: 2, 53996292670
2069: 1, 42168720310009: 1, 989231642989050: 1, 8772256481790: 1}
DS Math git:master #
```

and,

$$[3, 7]$$

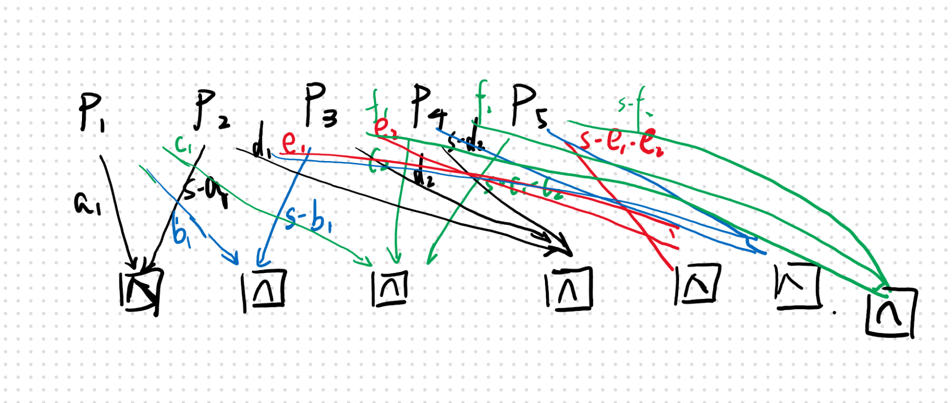
First we find $s = 516971327093293$ *, because it is the most seen number in the list of results.*

Than we find spi is $3, 7$ *because everytime when the answer is correct,* $3, 7$ *doesn't take any part.*

5. Let $= \{P1, P2, P3, P4, P5\}$. Design a secret sharing scheme that realizes an accessstructure with basis

$$\tau_0 = \{\{P_1, P_2\}, \{P_1, P_3\}, \{P_1, P_4, P_5\ \{P_2, P_3, P_4\}, \{P_2, P_3, P_5\}, \{P_2, P_4, P_5\}, \{P_3, P_4, P_5\}\}.$$

Solution:



p_1	a_1, b_1, c_1
p_2	$s - a_1, d_1, e_1, f_1$
p_3	$s - b_1, d_2, e_2, g_1$
p_4	$c_2, s - d_1 - d_2, f_2, g_2$
p_5	$s - c_1 - c_2, s - e_1 - e_2, s - f_1 - f_2, s - g_1 - g_2$

6. Let $\rho = \{P_1, P_2, ..., P_{20}\}$ be a set of 20 participants. Let $\tau = \{A : A \in P, |A| \geq 11\}$ be an access structure.

   (a) If τ is realized with the monotone circuit construction, how many numbers are there in theshare of each participant?

   (b) If τ is realized with $Shamir's(11, 20) - threshold$ secret sharing scheme, how many numbersare there in the share of each participant?

Solution:

(a) W.L.O.G. use P_1 as example.

The basis τ_0 of τ : $\{A|A \subseteq P, |A| = 11\}$

p_1 share is C_{19}^{10}

(b) each participant have only one share S_i for evert P_i, So, there is only one number in the share of each participant.