



Unit 26.

Hash Table

● Learning objectives

- ✓ ເຂົ້າໃຈບັນຫາ hashing ແລະ ສາມາດແກ້ໄຂບັນຫາການຄົ້ນຫາໂດຍໃຊ້ຕາຕະລາງ hash
- ✓ ສາມາດໃຊ້ໂຄງສ້າງຂໍ້ມູນວັດຈະນານຸກົມ(dictionary) ໃນພາສາ Python
- ✓ ສາມາດສ້າງຕາຕະລາງ hash ໃນພາສາ Python ແລະ ນຳໃຊ້ມັນສໍາລັບການແກ້ໄຂບັນຫາ

● Learning overview

- ✓ ຮຽນຮູ້ວິທີການຊ່ວຍບັນທຶກຄູ່ລະຫັດ-ຄ່າ ແລະ ທຳການຄົ້ນຫາໂດຍການນຳໃຊ້ໂຄງສ້າງຂໍ້ມູນ ວັດຈະນານຸກົມ
- ✓ ຮຽນຮູ້ວິທີການສ້າງຕາຕະລາງ hash ເຂົ້າໄປໃນ class ຂອງ Python
- ✓ ຮຽນຮູ້ວິທີການຄົ້ນຫາຂໍ້ມູນດ້ວຍເວລາຄົງທີ່ໂດຍໃຊ້ຕາຕະລາງ hash

● Concepts you will need to know from previous units

- ✓ ວິທີການບັນທຶກ ແລະ ຄົ້ນຫາຂໍ້ມູນໂດຍໃຊ້ຄ່າລະຫັດໂດຍຜ່ານວັດຈະນານຸກົມ
- ✓ ວິທີການກຳນົດຕົວປ່ຽນ ແລະ methods ໂດຍຜ່ານ class ແລະ ວິທີການ
- ✓ ວິທີການສະແດງເວລາການປະມວນຜົນຂອງຂັ້ນຕອນວິທີໂດຍໃຊ້ຕຳລາ Big-O

Keywords

Hash Table

Hashing

Hash Function

Dictionary

Constant Time

| Mission

1. Real world problem

1.1. ຕົວເລກໂຣມັນ (Roman numbers)

I	II	III	IV	V
1	2	3	4	5
VI	VII	VIII	IX	X
6	7	8	9	10
L	C	D	M	
50	100	500	1000	

- ▶ ໃນອະດີດ, ຕົວເລກໂຣມັນຖືກນຳໃຊ້ໃນ Rome ບູຮານ.
- ▶ ສ້າງຕົວປ່ຽນເລກໂຣມັນທີ່ໃຊ້ປ່ຽນແປງເລກໂຣມັນເປັນຕົວເລກອາຣັບ.

2. Mission

2.1.ການປ່ຽນເລກໂຣມັນ

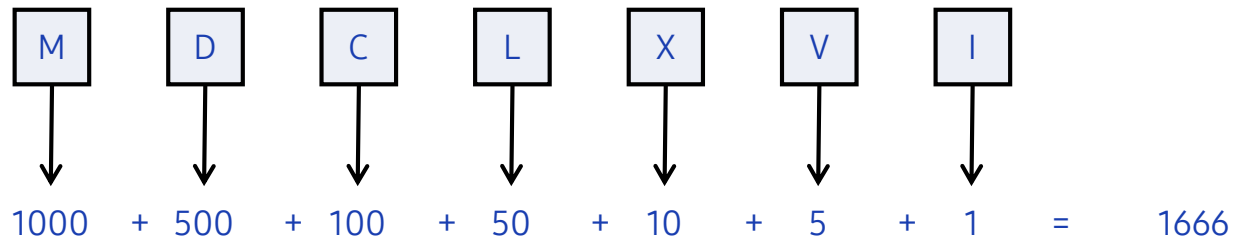
Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

- ▶ ຕົວເລກໂຣມັນສະແດງອອກດ້ວຍ 7 ສັນຍາລັກທີ່ແຕກຕ່າງກັນ.
- ▶ ຕາຕະລາງຢູ່ເບື້ອງຊ້າຍສະແດງໃຫ້ເຫັນຄ່າທີ່ແຕ່ລະສັນຍາລັກເປັນຕົວແທນ.
- ▶ ຕົວເລກໂຣມັນຖືກຄິດໄລ່ໄດ້ງ່າຍຖ້າເພີ່ມສັນຍາລັກຕາມລຳດັບ.
- ▶ ຕົວຢ່າງ, III ແມ່ນການເພີ່ມ 1 ເປັນສາມຄັ້ງ ແລະ XVI ແມ່ນການເພີ່ມ 10, 5 ແລະ 1.
- ▶ ຢ່າງໃດກໍຕາມ, ຖ້າສັນຍາລັກທີ່ເພີ່ມເຂົ້າທາງຫລັງແມ່ນຕົວເລກທີ່ໃຫຍ່ກວ່າ, ໃຫ້ຫັກຄ່າຂອງສັນຍາລັກຢູ່ທາງຫນ້າ.
- ▶ ຕົວຢ່າງ, IX ແມ່ນການບວກຂອງ 10 ແລະ ລົບ 1, ໃນຂະນະທີ່ IV ແມ່ນການບວກຂອງ 5 ແລະ ລົບ 1.

2Mission

2.1.ການປ່ຽນເລກໂຣມັນ

- ຈະເຮັດແນວໃດເພື່ອແປງຕົວເລກໂຣມັນ MDCLXVI?
- ດັ່ງທີ່ສະແດງຢູ່ຂ້າງລຸ່ມ, ປ່ຽນສັນຍາລັກແຕ່ລະອັນເປັນຕົວເລກອາຣັບແລະຫຼັງຈາກນັ້ນບວກພວກມັນເຂົ້າກັນ.



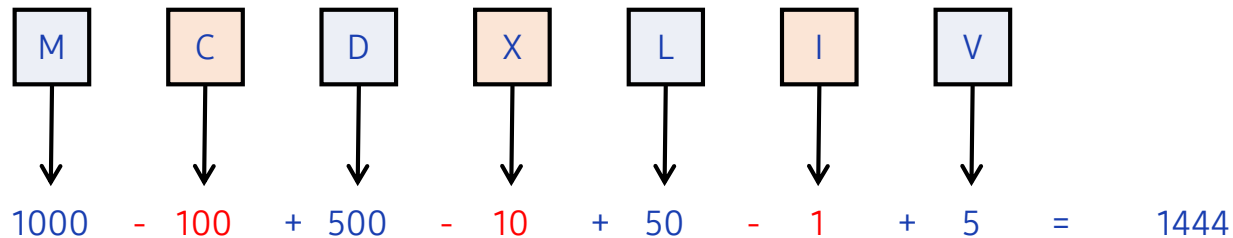
```
1 roman = input("Input a Roman number: ")
2 number = roman_to_int(roman)
3 print(number)
```

Input a Roman number: MDCLXVI
1666

2. Mission

2.1.ການປ່ຽນເລກໂຣມັນ

- | ຈະເຮັດແນວໃດເພື່ອປ່ຽນຕົວເລກໂຣມັນ MCDXLIV?
- | ໃນກໍລະນີນີ້, ເນື່ອງຈາກວ່າ C ຫນ້ອຍກວ່າ D, ທ່ານຈຳເປັນຕ້ອງລົບ C. ເຊັ່ນດຽວກັນ, ລົບ X ແລະ I ຈາກ XL ແລະ IV.



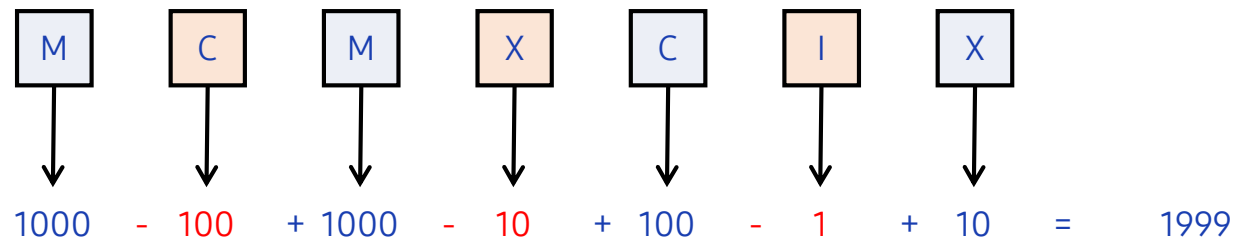
```
1 roman = input("Input a Roman number: ")
2 number = roman_to_int(roman)
3 print(number)
```

Input a Roman number: MCDXLIV
1444

2. Mission

2.1. ການປ່ຽນເລກໂຣມັນ

ນອກຈາກນັ້ນ, ສໍາລັບ MCMXCIX, ເນື່ອງຈາກວ່າຄ່າຂອງ C, X, I ມີຄ່ານ້ອຍກວ່າຄ່າຂອງ M, C, X ທີ່ຢູ່ທາງຫລັງ, ພວກມັນຈຳເປັນຕ້ອງຖືກລົບອອກ.



```
1 roman = input("Input a Roman number: ")
2 number = roman_to_int(roman)
3 print(number)
```

Input a Roman number: MCMXCIX
1999

3. Solution

3.1. ວິທີທີ່ຕົວປ່ຽນເລກໂຣມັນເຮັດວຽກ

ໂຄ້ດລຸ່ມນີ້ແມ່ນການປ່ຽນຕົວເລກໂຣມັນເປັນຕົວເລກອາຣັບ.

```
1 table = {'I':1, 'V':5, 'X':10, 'L':50, 'C':100, 'D':500, 'M':1000}
2 roman = input("Input a Roman number: ")
3 number = roman_to_int(roman)
4 print(number)
```

Input a Roman number: MCDXLIV
1444

3. Solution

3.1. ວິທີທີ່ໂຄ້ດປ່ຽນເລກໂຣມັນເຮັດວຽກ

```
1 def roman_to_int(str):
2     result = 0
3     for i in range(len(str) - 1):
4         if table[str[i]] < table[str[i + 1]]:
5             result -= table[str[i]]
6         else:
7             result += table[str[i]]
8     return result + table[str[-1]]
```

| Key concept

1. Hash Table

1.1. ຄຳນິຍາມຂອງ hash table

- | ຕາຕະລາງ hash, ຫຼື ແຜນທີ່ hash ໝາຍເຖິງປະເພດຂໍ້ມູນນາມມະທຳທີ່ເກັບຮັກສາ ແລະ ແຍກຂໍ້ມູນ ໂດຍການປ່ຽນແປງ key ໄປເປັນ value ໂດຍໃຊ້ການເຮັດ hashing.
- | Hashing ໝາຍເຖິງການກຳນົດທີ່ຢູ່ເພື່ອບັນທຶກຂໍ້ມູນຫຼືຊອກຫາຂໍ້ມູນທີ່ບັນທຶກໄວ້ໂດຍຄ່າຂໍ້ມູນ. ຟັງຊັນ Hash ກຳນົດລະຫັດໂດຍການນຳໃຊ້ຄ່າຂໍ້ມູນ.

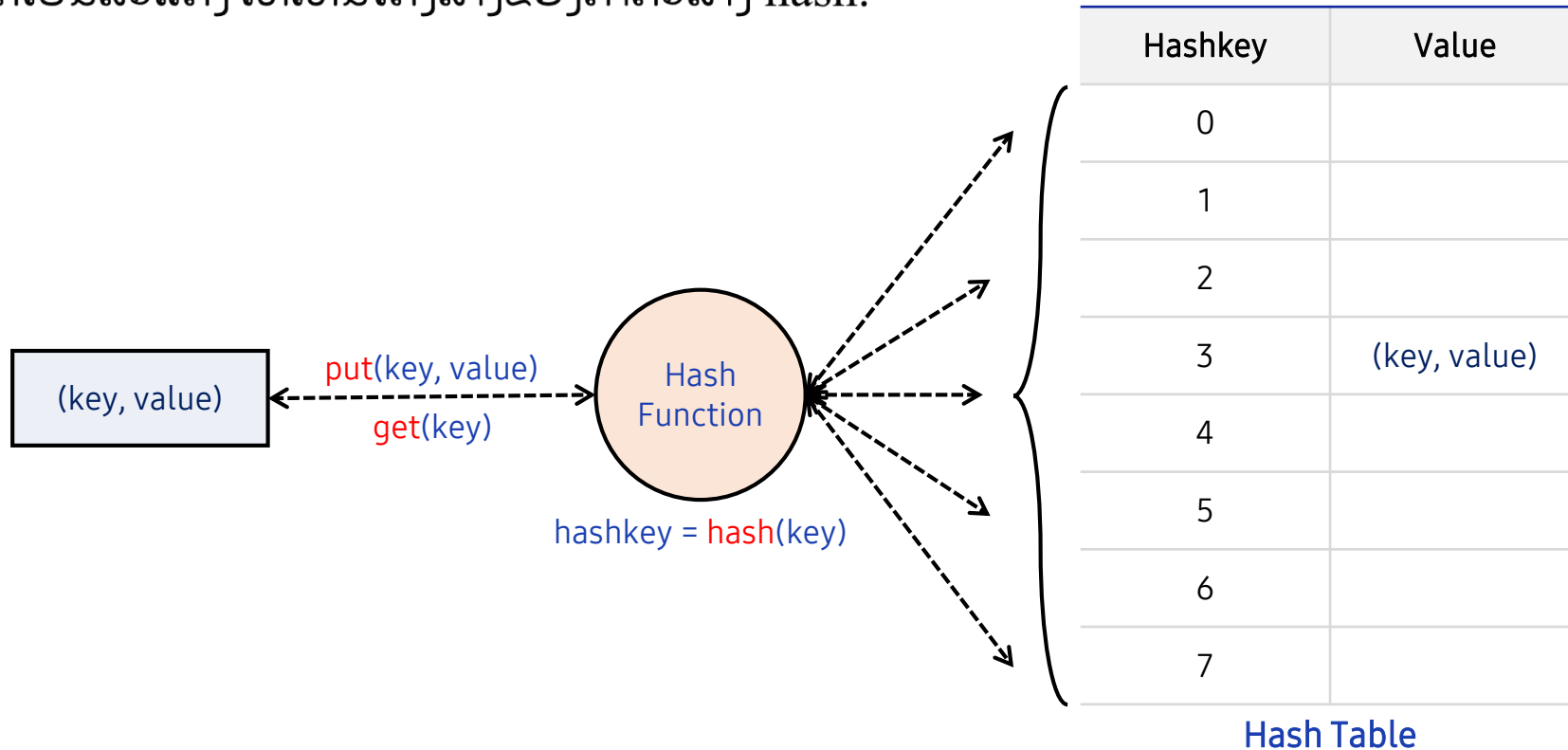
🔗 **Focus** ຕາຕະລາງ Hash ຈະໃຫ້ການໂຕ້ຕອບສາມຢ່າງທີ່ສຳຄັນດັ່ງຕໍ່ໄປນີ້.

- ▶ put(key, value): ເອົາຄ່າຂໍ້ມູນເຂົ້າໄປເກັບ ທີ່ຖືກກຳນົດໂດຍ key ແລະ hash value
- ▶ get(key): ເອີ້ນເອົາຄ່າທີ່ເກັບໄວ້ອອກມາໃຊ້ງານ ທີ່ກຳນົດໂດຍຄ່າ key
- ▶ hash(key): ປ່ຽນຄ່າ key ຂໍ້ມູນເປັນຄ່າ hash ໃນຊ່ວງໃດໜຶ່ງ

1. Hash Table

1.1. ຄຳນິຍາມຂອງ hash table

ໄຕ່ໄປນີ້ສະແດງໃຫ້ເຫັນໂຄງສ້າງຂອງຕາຕະລາງ hash.



1. Hash Table

1.1. ຕົວຢ່າງຂອງ hash table

❑ ຕົວຢ່າງເຊັ່ນ: ໃຫ້ຈົນຕະນາການວ່າເຈົ້າເອົາປຶ້ມ 5 ຫົວໄວ້ໃນຮ້ານວາງປຶ້ມ.

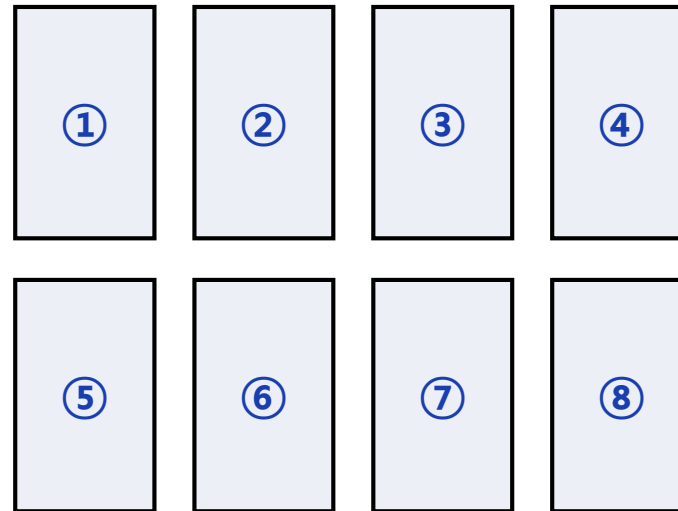
The Little Prince

The Old Man and the Sea

The Little Mermaid

Beauty and the Beast

The Last Leaf



Bookshelf with 8 slots

1. Hash Table

1.1. ຕົວຢ່າງຂອງ hash table

- | ກຳນົດຄ່າຫຼັກແບບສຸ່ມໃຫ້ກັບປຶ້ມແຕ່ລະຫົວ.
- | ໃນນີ້, ໃຊ້ຜົນບວກຂອງລະຫັດ ASCII ຂອງຫົວຂໍ້ປຶ້ມເປັນຄ່າລະຫັດຂອງປຶ້ມ.

Title T h e L a s t L e a f

ASCII Codes 84 + 104 + 101 + 32 + 76 + 97 + 115 + 116 + 32 + 76 + 101 + 97 + 102 = 1,133

```
1 list(map(ord, "The Last Leaf"))
```

```
[84, 104, 101, 32, 76, 97, 115, 116, 32, 76, 101, 97, 102]
```

1. Hash Table

1.1. ຕົວຢ່າງຂອງ hash table

■ ຄຳລະຫັດສຳລັບປຶ້ມແຕ່ລະຫົວສາມາດຖືກຄິດໄລ່ດ້ວຍ code ຕໍ່ໄປນີ້.

```
1 books = [  
2     "The Little Prince",  
3     "The Old Man and the Sea",  
4     "The Little Mermaid",  
5     "Beauty and the Beast",  
6     "The Last Leaf"  
7 ]  
8 for book in books:  
9     key = sum(map(ord, book))  
10    print(key, book)
```

```
1584 The Little Prince  
1929 The Old Man and the Sea  
1678 The Little Mermaid  
1837 Beauty and the Beast  
1133 The Last Leaf
```

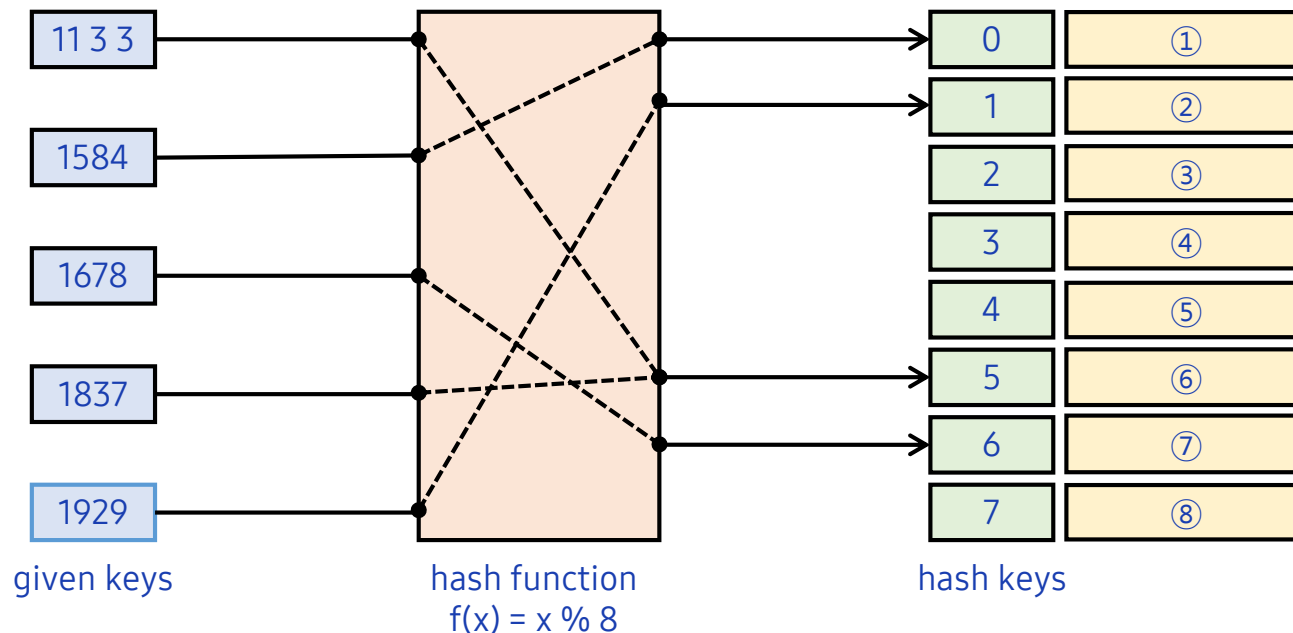
Line 8-10

- ຟັງຊັນ ord ສົ່ງຄືນຄ່າ ASCII Code ຂອງຕົວອັກສອນໃດໜຶ່ງ, ດັ່ງນັ້ນຈຶ່ງໄດ້ຮັບຄ່າ ASCII ຂອງແຕ່ລະຕົວອັກສອນ ໂດຍການໃຊ້ຟັງຊັນ map.
- ກຳນົດຜົນບວກຂອງລະຫັດ ASCII ຂອງຕົວອັກສອນທັງໝົດເປັນຄ່າລະຫັດຂອງປຶ້ມ.

1. Hash Table

1.3. Hashing ແລະ hash ຟັງຊັນ

- | Hashing ໝາຍເຖິງຂະບວນການປ່ຽນລະຫັດທີ່ໃຫ້ມາໄປເປັນລະຫັດອື່ນ ແລະ ຟັງຊັນ hash ໝາຍເຖິງຟັງຊັນທີ່ປ່ຽນຄວາມຍາວລະຫັດແບບສຸ່ມເປັນລະຫັດອື່ນທີ່ມີຄວາມຍາວຄົງທີ່. ລະຫັດທີ່ປ່ຽນໂດຍຟັງຊັນ hash ເອີ້ນວ່າ hash key.
- | ມີ 8 ຊ່ອງຢູ່ໃນຮ້ານວາງປຶ້ມ, ສະນັ້ນໃຫ້ກຳນົດຟັງຊັນ hash ເປັນ $f(x) = x \% 8$.



1. Hash Table

1.4. ກຳນົດ hash table ເປັນປະເພດ class

ການກຳນົດຕາຕະລາງ hash ເປັນປະເພດ class ມີດັ່ງນີ້.

```
1 class HashTable:
2
3     def __init__(self, size):
4         self.size = size
5         self.table = {}
6         for i in range(size):
7             self.table[i] = []
8
9     def hash(self, key):
10        return key % self.size
11
12    def get(self, key):
13        return self.table[self.hash(key)]
14
15    def put(self, key, value):
16        bucket = self.table[self.hash(key)]
17        if value not in bucket:
18            bucket.append(value)
```

1. Hash Table

1.4. ກຳນົດ hash table ເປັນປະເພດ class

Hash table constructor ເລີ່ມຕົ້ນຕາຕະລາງ hash ຕາມຂະໜາດຂອງລະຫັດທີ່ໃຫ້.

```
1 class HashTable:
2
3     def __init__(self, size):
4         self.size = size
5         self.table = {}
6         for i in range(size):
7             self.table[i] = []
```



Line 3-7

- 'size' ເປັນພາຣາມິເຕີຂໍ້ມູນປ້ອນເຂົ້າແມ່ນຂະໜາດຂອງຕາຕະລາງ hash ແລະຕາຕະລາງ hash ຖືກກຳນົດໂດຍ dictionary ຂອງ Python.
- ໃນ dictionary, ເລີ່ມຕົ້ນຖັງໃສ່ຈຳນວນຂະໜາດ. ທຳອິດ, ຖັງກາຍເປັນ list ເປົ່າ.

1. Hash Table

1.4. ກຳນົດ hash table ເປັນປະເພດ class

ໃນຕາຕະລາງ hash, ກຳນົດຟັງຊັນ hash ເປັນເປັນຟັງຊັນປະຕິບັດການເລກຄະນິດສາດແບບໂມດູນຕາມຂະໜາດຂອງລະຫັດທີ່ໃຫ້.

```
1 class HashTable:
2
3     def __init__(self, size):
4         self.size = size
5         self.table = {}
6         for i in range(size):
7             self.table[i] = []
8
9     def hash(self, key):
10         return key % self.size
11
```

Line 3-7

- 'size' ເປັນພາຣາມິເຕີຂໍ້ມູນປ້ອນເຂົ້າແມ່ນຂະໜາດຂອງຕາຕະລາງ hash ແລະຕາຕະລາງ hash ຖືກກຳນົດໂດຍ dictionary ຂອງ Python.
- ໃນ dictionary, ເລີ່ມຕົ້ນຖັງໃສ່ຈຳນວນຂະໜາດ. ທຳອິດ, ຖັງກາຍເປັນ list ເປົ່າ.

1. Hash Table

1.4. ກຳນົດ hash table ເປັນປະເພດ class

Method put() ໄດ້ຮັບລະຫັດ ແລະ ຄ່າເປັນຂໍ້ມູນປ້ອນເຂົ້າ ແລະ ເກັບຄ່າໄວ້ໃນຖັງໃສ່ລະຫັດ hash.

```
11  
12     def get(self, key):  
13         return self.table[self.hash(key)]  
14  
15     def put(self, key, value):  
16         bucket = self.table[self.hash(key)]  
17         if value not in bucket:  
18             bucket.append(value)
```

Line 15-18

- ຊອກຫາຖັງທີ່ຄ່າລະຫັດແມ່ນ self.hash(key) ໃນວັດຈະນານຸກົມ self.table.
- ເມື່ອບໍ່ອະນຸຍາດໃຫ້ເກັບຄ່າຊ້າກັນ, ໃຫ້ເພີ່ມຄ່າໃນຖັງເກັບຂໍ້ມູນໄດ້ ຖ້າຖັງບໍ່ມີຄ່າ.

1. Hash Table

1.4. ກຳນົດ hash table ເປັນປະເພດ class

Method `get()` ໄດ້ຮັບລະຫັດເປັນ ຂໍ້ມູນປ້ອນເຂົ້າ ແລະ ສົ່ງຄືນ ລະຫັດ hash ຂອງຖັງເກັບຂໍ້ມູນ.

```
11  
12     def get(self, key):  
13         return self.table[self.hash(key)]  
14  
15     def put(self, key, value):  
16         bucket = self.table[self.hash(key)]  
17         if value not in bucket:  
18             bucket.append(value)
```



Line 12-13

- Method `get()` ໄດ້ຮັບລະຫັດເປັນ ຂໍ້ມູນປ້ອນເຂົ້າ ແລະ ສົ່ງຄືນ ລະຫັດ hash ຂອງຖັງເກັບຂໍ້ມູນ.

1. Hash Table

1.5. ຕົວຢ່າງການໃຊ້ງານຕາຕະລາງ hash

ເລີ່ມຕົ້ນຕາຕະລາງ hash ທີ່ຖືກກໍານົດໄວ້ກ່ອນຫນ້ານີ້ເຂົ້າໄປໃນຕາຕະລາງທີ່ມີ 8 ຖັງ.

```
1 table = HashTable(8)
2 for key in table.table.keys():
3     print(key, table.table[key])
```

```
0 []
1 []
2 []
3 []
4 []
5 []
6 []
7 []
```

1. Hash Table

1.5. ຕົວຢ່າງການໃຊ້ງານຕາຕະລາງ hash

ເພີ່ມປຶ້ມທີ່ຊື່ " The Little Prince" ໃນຕາຕະລາງ hash ທີ່ສ້າງຂຶ້ນ, ແລະມັນຈະຖືກເກັບໄວ້ໃນຖັງທີ 0.

```
1 table = HashTable(8)
2 book = "The Little Prince"
3 key = sum(map(ord, book))
4 table.put(key, book)
5 for key in table.table.keys():
6     print(key, table.table[key])
```

```
0 ['The Little Prince']
1 []
2 []
3 []
4 []
5 []
6 []
7 []
```

1. Hash Table

1.5. ຕົວຢ່າງການໃຊ້ງານຕາຕະລາງ hash

I ເອົາປຶ້ມ 4 ຫົວທີ່ຍັງເຫຼືອໃສ່ໃນຮ້ານວາງປຶ້ມດັ່ງນີ້.

```
1 books = [  
2     "The Old Man and the Sea",  
3     "The Little Mermaid",  
4     "Beauty and the Beast",  
5     "The Last Leaf"  
6 ]  
7 for book in books:  
8     key = sum(map(ord, book))  
9     table.put(key, book)  
10 for key in table.table.keys():  
11     print(key, table.table[key])
```

```
0 ['The Little Prince']  
1 ['The Old Man and the Sea']  
2 []  
3 []  
4 []  
5 ['Beauty and the Beast', 'The Last Leaf']  
6 ['The Little Mermaid']  
7 []
```

1. Hash Table

1.5. ຕົວຢ່າງການໃຊ້ງານຕາຕະລາງ hash

❖ ເພື່ອຊອກຫາຊ່ອງເກັບບ່ອນທີ່ປຶ້ມ ຊື່ “The Last Leaf” ຢູ່, ໃຫ້ໃຊ້ method `get()`.

```
1 title = "The Last Leaf"  
2 key = sum(map(ord, title))  
3 bucket = table.get(key)  
4 print(key, bucket)
```

```
1133 ['Beauty and the Beast', 'The Last Leaf']
```

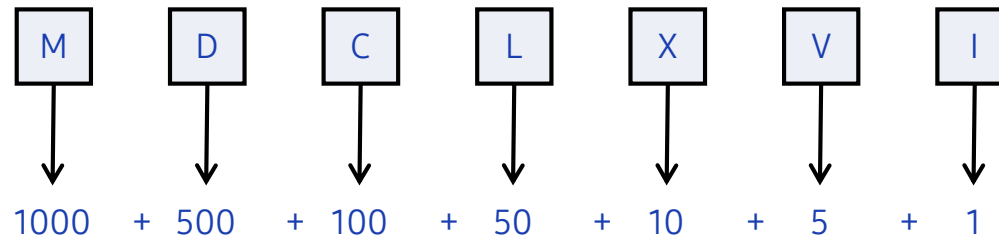
| Let's code

1. ໂປຣແກຣມປ່ຽນເລກໂຣມັນ

1.1. ກຳນົດຕາຕະລາງ hash ໂດຍໃຊ້ປະເພດ dictionary

ລອງເບິ່ງໂປຣແກຣມປ່ຽນຕົວເລກໂຣມັນ. Dictionary ຂອງ Python ສາມາດຖືກວ່າເປັນຕາຕະລາງ hash ຫຼື hash map ທີ່ປ່ຽນຕົວເລກໂຣມັນເປັນຕົວເລກອາຣັບໂດຍໃຊ້ລະຫັດ hash ຂອງຕົວອັກສອນ.

```
1 table = {'I':1, 'V':5, 'X':10, 'L':50, 'C':100, 'D':500, 'M':1000}
```



1. ໂປຣແກຣມປ່ຽນເລກໂຣມັນ

1.2. ສ້າງ ແລະ ຂຽນໂປຣແກຣມ

I ການແປງຕົວເລກໂຣມັນຖືກສ້າງດັ່ງຕໍ່ໄປນີ້ໂດຍໃຊ້ຕາຕະລາງ hash.

```
1 def roman_to_int(str):
2     result = 0
3     for i in range(len(str) - 1):
4         if table[str[i]] < table[str[i + 1]]:
5             result -= table[str[i]]
6         else:
7             result += table[str[i]]
8     return result + table[str[-1]]
```

Line 1-3

- ຟັງຊັນ `roman_to_int()` ປ່ຽນ `str`, ເຊິ່ງເປັນພາຣາມິເຕີປ້ອນເຂົ້າ, ໄປເປັນຕົວເລກອາຣັບ.
- ຄ່າຜົນໄດ້ຮັບແມ່ນເລີ່ມຕົ້ນເປັນ 0 ໃນຕອນທຳອິດ. ປະຕິບັດການບວກຫຼືລົບກັບຕົວອັກສອນທັງຫມົດ.

1. ໂປຣແກຣມປ່ຽນເລກໂຣມັນ

1.2. ສ້າງ ແລະ ຂຽນໂປຣແກຣມ

```
1 def roman_to_int(str):
2     result = 0
3     for i in range(len(str) - 1):
4         if table[str[i]] < table[str[i + 1]]:
5             result -= table[str[i]]
6         else:
7             result += table[str[i]]
8     return result + table[str[-1]]
```



Line 4-7

- ຖ້າຄ່າຕົວອັກສອນທີ່ຢູ່ເບື້ອງໜຶ່ງໃຫຍ່ກວ່າຄ່າຕົວອັກສອນປັດຈຸບັນ, ໃຫ້ລົບຄ່າອອກຈາກຄ່າປັດຈຸບັນ.
- ຖ້າບໍ່ແມ່ນດັ່ງນັ້ນ, ໃຫ້ເພີ່ມຄ່າຕົວອັກສອນໃສ່ຄ່າປັດຈຸບັນ.

1. ໂປຣແກຣມປ່ຽນເລກໂຣມັນ

1.2. ສ້າງ ແລະ ຂຽນໂປຣແກຣມ

```
1 def roman_to_int(str):
2     result = 0
3     for i in range(len(str) - 1):
4         if table[str[i]] < table[str[i + 1]]:
5             result -= table[str[i]]
6         else:
7             result += table[str[i]]
8     return result + table[str[-1]]
```

Line 8

- ເມື່ອອອກຈາກ for-loop, ຕົວອັກສອນທັງໝົດຍົກເວັ້ນຕົວສຸດທ້າຍຈະຖືກເພີ່ມ ຫຼື ລົບອອກ.
- ດັ່ງນັ້ນ, ໃຫ້ສິ່ງຄືນຜົນລວມຂອງຄ່າຂອງ str[-1] (ຕົວອັກສອນສຸດທ້າຍ) ແລະ ຜົນໄດ້ຮັບ.

| Pop quiz

Quiz. #1

| ໃຊ້ຟັງຊັນ hash ຂອງຕາຕະລາງ hash ເພື່ອຄິດໄລ່ລະຫັດ ແລະ hashkey ຂອງ "Alice in Wonderland."

```
1 table = HashTable(8)
2 book = "Alice in Wonderland"
3 key = sum(map(ord, book))
4 print(key, table.hash(key))
```

Quiz. #2

I ຖ້າມີ 10 ຊ່ອງຢູ່ໃນຮ້ານວາງປຶ້ມ, ຈົ່ງຄຳນວນດ້ວຍລະຫັດຕໍ່ໄປນີ້ເພື່ອຊອກຫາຊ່ອງທີ່ປຶ້ມແຕ່ລະຫົວຕໍ່ໄປນີ້ຈະຖືກໃສ່ເຂົ້າໄປ.

```
1 table = HashTable(10)
2 books = [
3     "The Little Prince",
4     "The Old Man and the Sea",
5     "The Little Mermaid",
6     "Beauty and the Beast",
7     "The Last Leaf",
8     "Alice in Wonderland"
9 ]
10 for book in books:
11     key = sum(map(ord, book))
12     table.put(key, book)
13 for key in table.table.keys():
14     print(key, table.table[key])
```

| Pair programming



Pair Programming Practice

| ແນວທາງ, ກົນໄກ ແລະ ແຜນສຸກເສີນ

ການກະກຽມການສ້າງໂປຣແກຣມຮ່ວມກັນເປັນຄູ່ກ່ຽວຂ້ອງກັບການໃຫ້ຄໍາແນະນໍາແລະກົນໄກເພື່ອຊ່ວຍໃຫ້ນັກຮຽນຈັບຄູ່ຢ່າງຖືກຕ້ອງແລະ ໃຫ້ເຂົາເຈົ້າເຮັດວຽກເປັນຄູ່. ຕົວຢ່າງ, ນັກຮຽນຄວນປຸງ "ເຮັດ." ການກະກຽມທີ່ມີປະສິດຕິຜົນຕ້ອງໃຫ້ມີແຜນການສຸກເສີນໃນກໍລະນີທີ່ຄູ່ຮ່ວມງານຫນຶ່ງບໍ່ຢູ່ທີ່ຕັດສິນໃຈທີ່ຈະບໍ່ເຂົ້າຮ່ວມດ້ວຍເຫດຜົນໃດຫນຶ່ງ ທີ່ດ້ວຍເຫດຜົນອື່ນ. ໃນກໍລະນີເຫຼົ່ານີ້, ມັນເປັນສິ່ງສໍາຄັນທີ່ຈະເຮັດໃຫ້ມັນຊັດເຈນວ່ານັກຮຽນທີ່ມີປະຕິບັດໜ້າທີ່ຢ່າງຫ້າວຫັນຈະບໍ່ຖືກລົງໂທດຍ້ອນວ່າການຈັບຄູ່ບໍ່ໄດ້ຜິດ.

| ການຈັບຄູ່ທີ່ຄ້າຍຄືກັນ, ບໍ່ຈໍາເປັນຕ້ອງເທົ່າທຽມກັນ, ຄວາມສາມາດເປັນຄູ່ຮ່ວມງານ

ການຂຽນໂປຣແກຣມຄູ່ຈະມີປະສິດທິພາບເມື່ອນັກຮຽນຕັ້ງໃຈຮ່ວມກັນເຮັດວຽກ, ຊຶ່ງວ່າມັນຈໍາເປັນຕ້ອງມີຄວາມຮູ້ເທົ່າທຽມກັນ, ແຕ່ຕ້ອງມີຄວາມສາມາດເຮັດວຽກເປັນຄູ່ຮ່ວມງານ. ການຈັບຄູ່ນັກຮຽນທີ່ບໍ່ສາມາດເຂົ້າກັນໄດ້ມັກຈະເຮັດໃຫ້ການມີສ່ວນຮ່ວມທີ່ບໍ່ສົມດຸນກັນ. ຄູ່ສອນຕ້ອງເນັ້ນຫນັກວ່າການຂຽນໂປຣແກຣມຄູ່ບໍ່ແມ່ນຍຸດທະສາດ “divide-and-conquer”, ແຕ່ຈະເປັນຄວາມພະຍາຍາມຮ່ວມມືເຮັດວຽກທີ່ແທ້ຈິງໃນທຸກໆດ້ານສໍາລັບໂຄງການທັງຫມົດ. ຄູ່ຄວນຫຼີກເວັ້ນການຈັບຄູ່ນັກຮຽນທີ່ອ່ອນຫຼາຍກັບນັກຮຽນທີ່ເກັ່ງຫຼາຍ.

| ກະຕຸ້ນນັກຮຽນໂດຍການໃຫ້ສິ່ງຈູງໃຈພິເສດ

ການສະເໜີແຮງຈູງໃຈພິເສດສາມາດຊ່ວຍກະຕຸ້ນນັກຮຽນໃຫ້ຈັບຄູ່, ໂດຍສະເພາະກັບນັກຮຽນທີ່ມີຄວາມສາມາດຫຼາຍ. ຈະເຫັນວ່າມັນເປັນປະໂຫຍດທີ່ຈະໃຫ້ນັກຮຽນຈັບຄູ່ເຮັດວຽກຮ່ວມກັນພຽງແຕ່ຫນຶ່ງຫຼືສອງວຽກເທົ່ານັ້ນ.



Pair Programming Practice

I ປ້ອງກັນການໂກງໃນການເຮັດວຽກຮ່ວມກັນ

ສິ່ງທ້າທາຍສໍາລັບຄູແມ່ນເພື່ອຊອກຫາວິທີທີ່ຈະປະເມີນຜົນໄດ້ຮັບຂອງບຸກຄົນ, ໃນຂະນະທີ່ນໍາໃຊ້ຜົນປະໂຫຍດຂອງການຮ່ວມມື. ຈະຮູ້ໄດ້ແນວໃດວ່ານັກຮຽນຕັ້ງໃຈເຮັດວຽກ ຫຼື ກົງແຮງງານຜູ້ຮ່ວມງານ? ຜູ້ຊ່ຽວຊານແນະນໍາໃຫ້ທົບທວນຄືນການອອກແບບບັນຫາສູດ ແລະ ການປະເມີນ ພ້ອມທັງປຶກສາຫາລືຢ່າງຈະແຈ້ງ ແລະ ຊັດເຈນກ່ຽວກັບພຶດຕິກຳຂອງນັກຮຽນທີ່ຈະຖືກຕິດຕາມວ່າຂໍ້ຕົວະ. ຜູ້ຊ່ຽວຊານເນັ້ນໜັກໃຫ້ຄູເຮັດການມອບໝາຍໃຫ້ມີຄວາມໝາຍຕໍ່ນັກຮຽນ ແລະ ອະທິບາຍຄຸນຄ່າຂອງສິ່ງທີ່ນັກຮຽນຈະຮຽນຮູ້ໂດຍການເຮັດສໍາເລັດ.

I ສະພາບແວດລ້ອມການຮຽນຮູ້ໃນການຮ່ວມມື

ສະພາບແວດລ້ອມການຮຽນຮູ້ໃນຮ່ວມກັນເກີດຂຶ້ນໄດ້ທຸກເວລາທີ່ຜູ້ສອນຮຽກຮ້ອງໃຫ້ນັກຮຽນເຮັດວຽກຮ່ວມກັນໃນກິດຈະກຳການຮຽນຮູ້. ສະພາບແວດລ້ອມການຮຽນຮູ້ຮ່ວມກັນສາມາດມີສ່ວນຮ່ວມທັງກິດຈະກຳທີ່ເປັນທາງການ ແລະ ບໍ່ເປັນທາງການ ແລະ ອາດຈະບໍ່ລວມເຖິງການປະເມີນໂດຍກົງ. ຕົວຢ່າງ, ນັກສຶກສາຄູ່ເຮັດວຽກມອບໝາຍຮ່ວມກັນໃນການຂຽນໂປຣແກຣມ; ນັກສຶກສາກຸ່ມນ້ອຍໆສິນທະນາຄໍາຕອບທີ່ເປັນໄປໄດ້ຕໍ່ກັບຄໍາຖາມຂອງອາຈານໃນລະຫວ່າງການບັນຍາຍ; ແລະ ນັກຮຽນເຮັດວຽກຮ່ວມກັນນອກຫ້ອງຮຽນເພື່ອຮຽນຮູ້ແນວຄວາມຄິດໃໝ່. ການຮຽນຮູ້ການຮ່ວມມືແມ່ນແຕກຕ່າງຈາກໂຄງການທີ່ນັກຮຽນ “divide and conquer.” ເມື່ອນັກຮຽນແບ່ງວຽກກັນ, ແຕ່ລະຄົນຮັບຜິດຊອບພຽງແຕ່ສ່ວນໜຶ່ງຂອງການແກ້ໄຂບັນຫາ ແລະ ຈະບໍ່ຄ່ອຍມີບັນຫາຫຍັງໃນການເຮັດວຽກຮ່ວມກັບຄົນອື່ນໃນທີມ. ໃນສະພາບແວດລ້ອມການເຮັດວຽກຮ່ວມກັນ, ນັກຮຽນມີສ່ວນຮ່ວມໃນການສິນທະນາປຶກສາຫາລືເຊິ່ງກັນແລະກັນ.

Q1. ສ້າງຕົວແປງທີ່ແປງຕົວເລກອາຫລັບເປັນຕົວເລກໂຣມັນໂດຍໃຊ້ຕາຕະລາງ hash ທີ່ສະໜອງໃຫ້ຂ້າງລຸ່ມນີ້.

```
1 table = {1000:'M', 900:'CM', 500:'D', 400:'CD',  
2          100:'C', 90:'XC', 50:'L', 40:'XL',  
3          10:'X', 9:'IX', 5:'V', 4:'XI', 1:'I'}
```

```
1 num = int(input("Input a number: "))  
2 print(int_to_roman(num))
```

Input a number: 1999
MCMXCIX

Examples of roman numbers:

369, 80, 29, 155, 14, 492, 348, 301, 469, 499

CCCLXIX, LXXX, XXIX, CLV, XIV, CDXCII, CCCXLVIII, CCCI, CDLXIX, CDXCIX,



| End of Document



SAMSUNG

Together for Tomorrow!
Enabling People

Education for Future Generations

©2021 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of book.

This book is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this book other than the curriculum of Samsung innovation Campus or to use the entire or part of this book, you must receive written consent from copyright holder.

