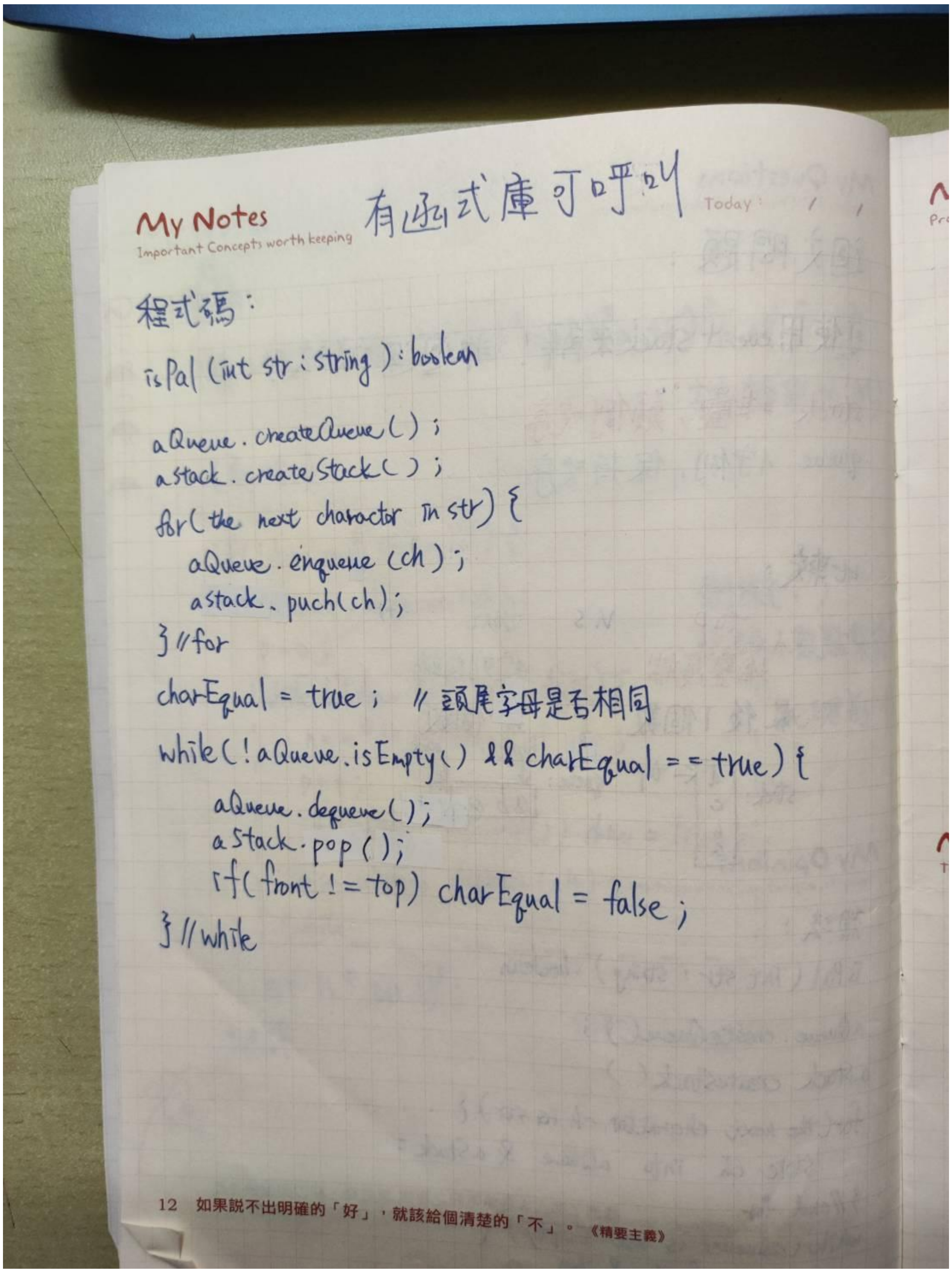


My Notes
Important Concepts worth keeping

有函式庫可呼叫

Today:　/　/

程式碼：

```
isPal (int str : string) : boolean

aQueue.createQueue( );
aStack.createStack( );
for( the next charactor in str) {
    aQueue.enqueue (ch);
    aStack.puch(ch);
} //for
charEqual = true;  // 頭尾字母是否相同
while(!aQueue.isEmpty( ) && charEqual == true) {
    aQueue.dequeue( );
    aStack.pop ( );
    if( front != top) charEqual = false;
} // while
```

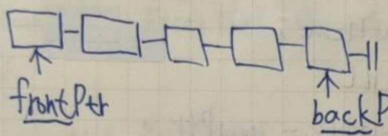
My Questions
Problems & Difficulties needing exploration

Implementation of ADT Queue
(實作佇列的抽象資料類別)

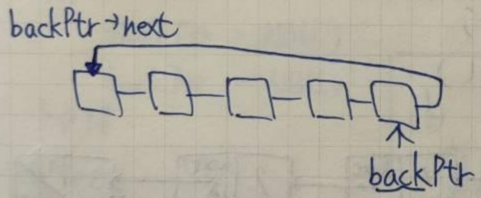
My learning weather report

兩種方法：

① A linear linked list 鏈狀



② A circular linked list 环狀



My Opinions
Thoughts, inspirations, and suggestions

```
class Queue {

  public :
        bool  isEmpty( );
        void  enqueue( const QueueItemType & newItem);
        void  dequeue( );
        void  dequeue(QueueItemType& queueFront);
        void  getFront (QueueItemTyp & queueFront) const;
  private : struct QueueNode {
            QueueItemType  item;
            QueueNode     *next;
  };                                    QueueNode *backPtr;
                                    →   QueueNode *FrontPtr;
```

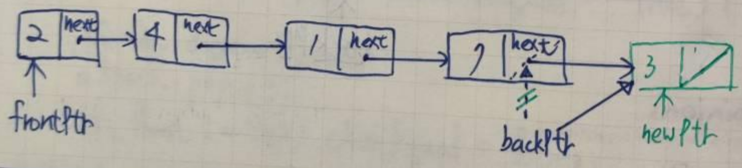
密碼 cipher key

If it isn't a clear yes, then it's a clear no. 《Essentialism》

鏈狀 Queue 實作

enqueue

```
void Queue :: equeue ( const QueueItemType & newItem){
QueueNode* newPtr = new QueueNode;
newPtr→Item = newItem;
newPtr→next = NULL;
if(isEmpty()) frontPtr = newPtr;
else backPtr→next= newPtr;
backPtr = newPtr;
}// Queue::enqueue()
```

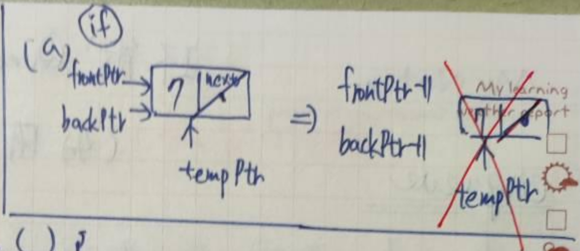


get Front

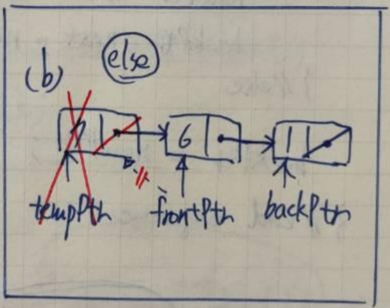
```
void Queue :: getFront( QueueItemType & queueFront){
  if(isEmpty()) throw QueueException(" ..... ");
  else queueFront = frontPtr → item;
}// Queue :: getFront()
```

My Questions
Problems & Difficulties needing exploration

dequeue



```
void Queue : dequeue ( ) {
    if ( isEmpty ( ))
        throw QueueException ( "Queue Exeception ..... " ) ;
    else {
      QueueNode *tempPtr = new frontPtr;
      if ( frontPtr == backPtr ) {
          frontPtr = NULL;
          backPtr = NULL;
      } // if
      else {
```



My Opinions
Thoughts, inspirations and suggestions

```
          front = front → next ;
      } // else
      tempPtr → next = NULL ;
      delete tempPtr ;
    } // else
} // Queue : dequeue ( )
```

密碼
cipher key

环狀 Queue 實作

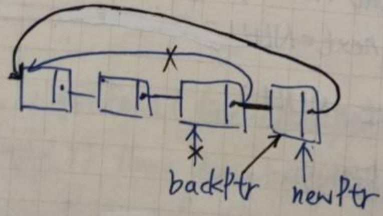
(好用!)

Today: / /

enqueue

```
void Queue::enqueue ( const QueueItemType & newItem){
  QueueNode * newPter = new QueueNode;
  newPtr -> Item = newItem;
  if( IsEmpty())
       newPtr -> next = newPtr;
  else {
       newPtr -> next = backPtr -> next;
       backPtr -> next = newPtr;
  } //else

  backPtr = newPtr;
} //end enqueue
```



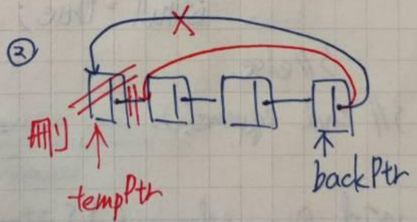
My Questions
Problems & Difficulties needing exploration

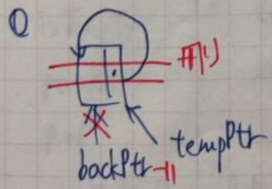
My learning weather report

dequeue : dequeue 和 getfrom 的合併版

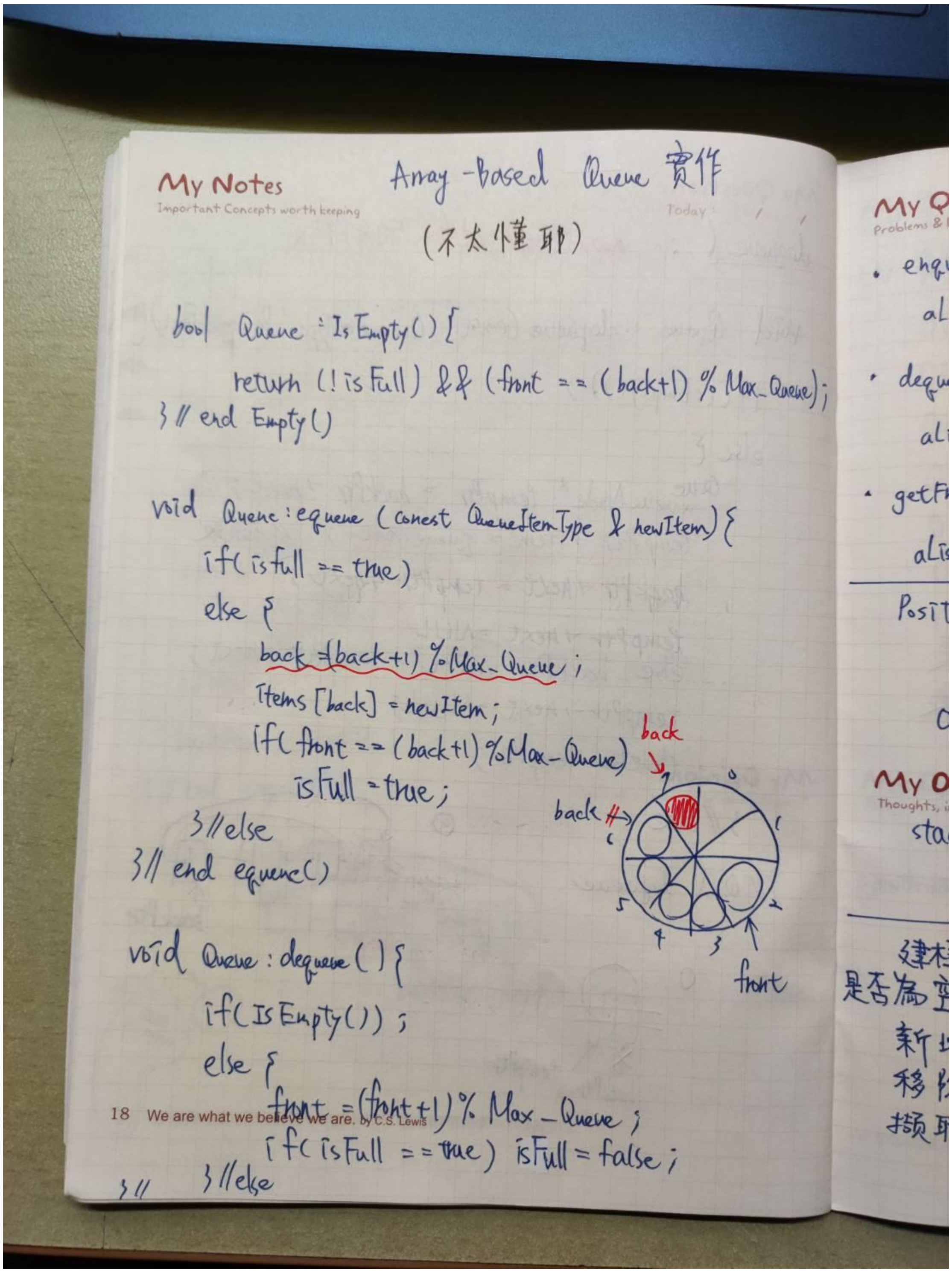
```
void Queue::dequeue(const QueueItemType & queueFront){
   if(isEmpty());
   else {
      QueueNode* tempPtr = backPtr→next;
      tempPtr→item = queueFront;  // 擷取
      if(backPtr == backPtr→next)
           backPtr = NULL;
      else  backPtr→next = tempPtr→next;
      tempPtr→next = NULL;
      delete tempPtr;
   }// else
}// end dequeue
```

My Opinions
- Thoughts, inspirations, and suggestions





密碼 cipher key



My Notes
Important Concepts worth keeping

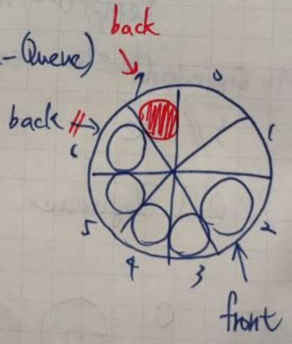
Array-Based Queue 實作
(不太懂耶)

Today / /

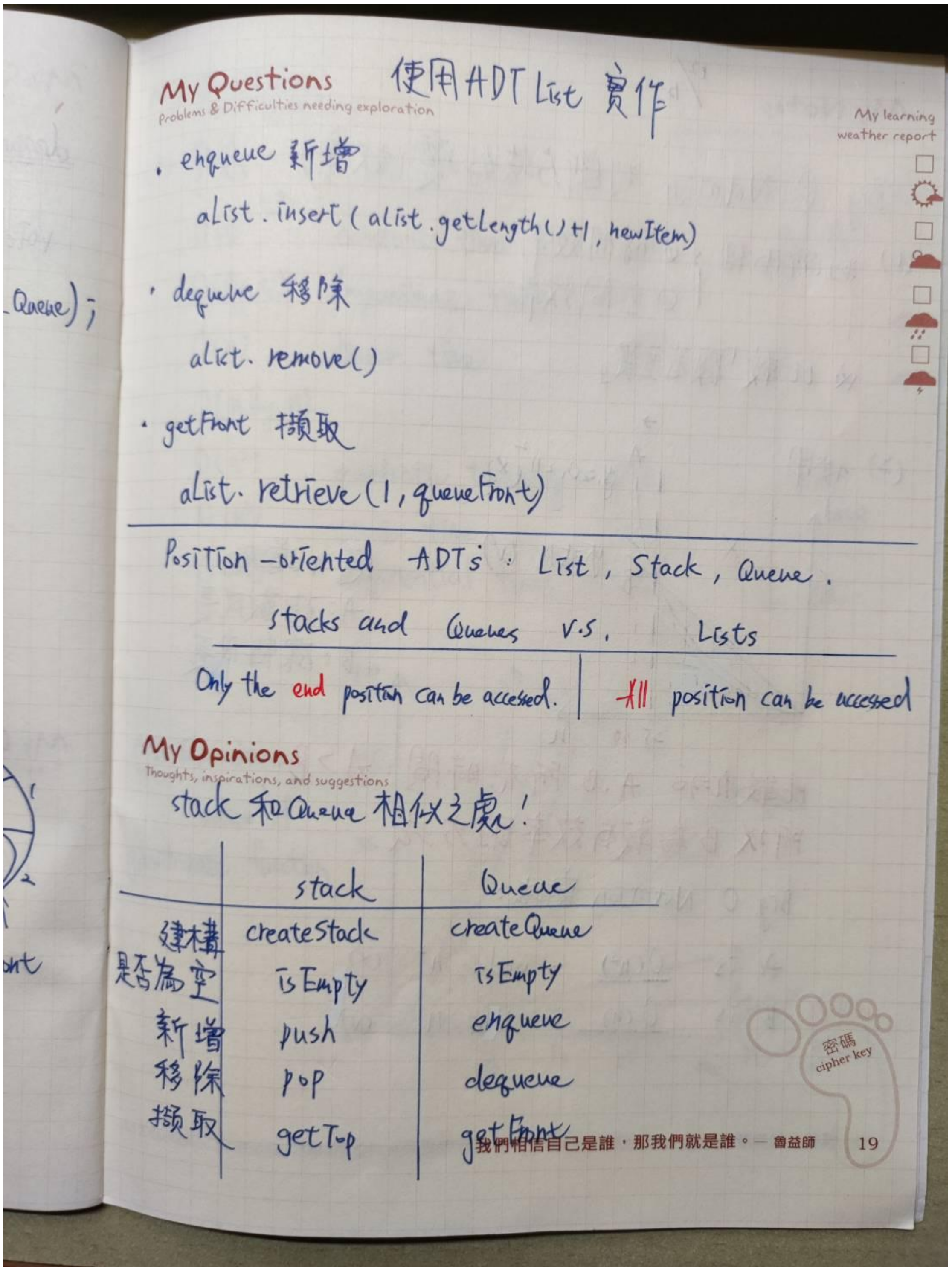
```
bool Queue::IsEmpty(){
    return (!isFull) && (front == (back+1) % Max_Queue);
}// end Empty()

void Queue::equeue (const QueueItemType & newItem){
  if(isfull == true)
  else {
    back =(back+1) % Max_Queue ;
    Items [back] = newItem;
    if( front == (back+1) %Max-Queue)
        isFull = true;
  }//else
}// end equeue()

void Queue::dequeue(){
  if(IsEmpty());
  else {
    front =(front+1) % Max_Queue;
    if( isFull == true) isFull = false;
}//  }//else
```



My Q
Problems &
• enq
al
• dequ
al
• getF
alis
Posit
C
My O
Thoughts,
sta
建
是否為空
新
移
擷



My Questions 使用ADT List 實作

Problems & Difficulties needing exploration

My learning weather report

- enqueue 新增

  aList.insert(aList.getLength()+1, newItem)

- dequeue 移除

  aList.remove()

- getFront 擷取

  aList.retrieve(1, queueFront)

Position-oriented ADTs : List, Stack, Queue.

Stacks and Queues v.s.	Lists
Only the end position can be accessed.	All position can be accessed

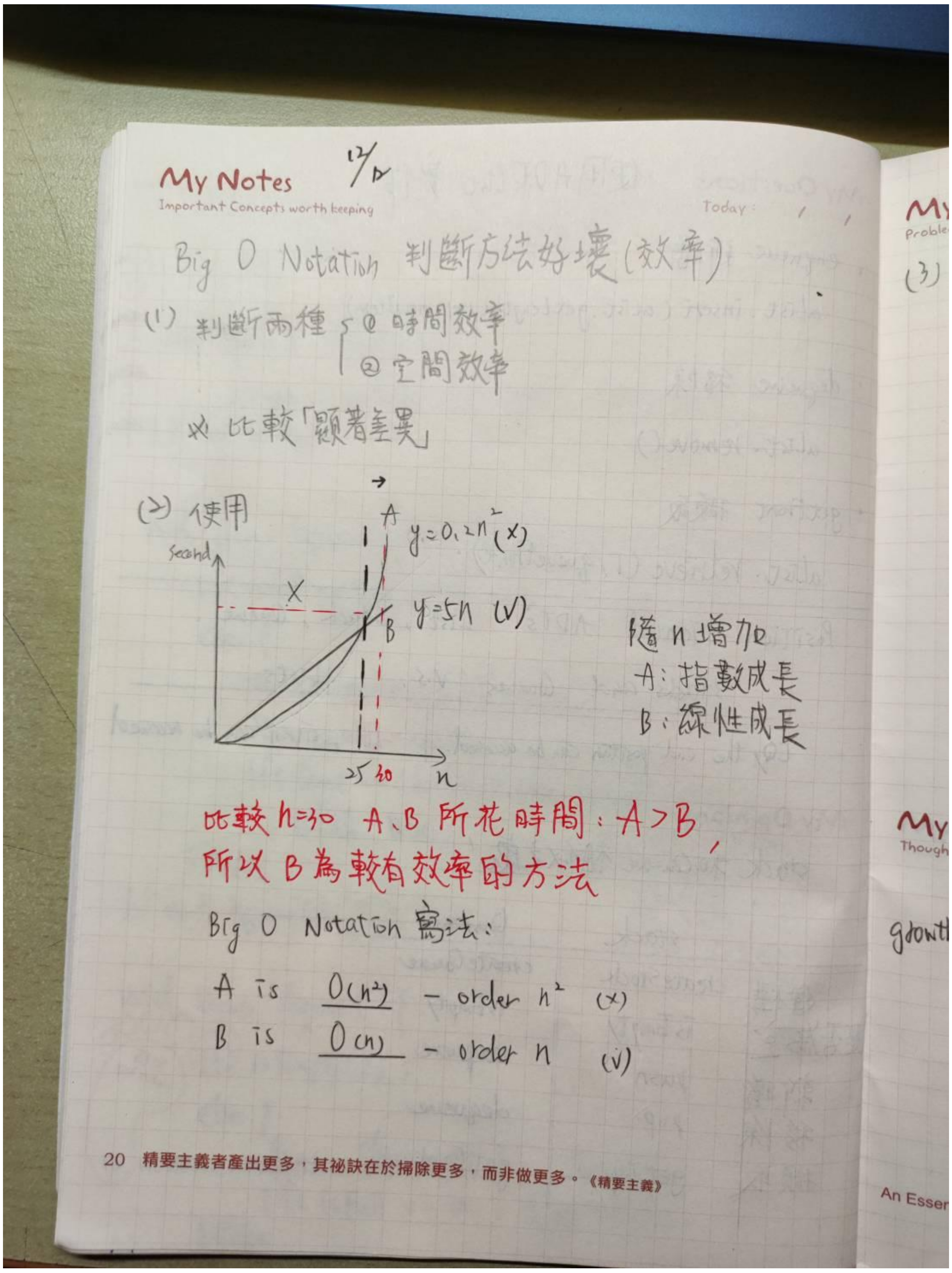
My Opinions

Thoughts, inspirations, and suggestions

stack 和 Queue 相似之處!

	stack	Queue
建構	createStack	createQueue
是否為空	isEmpty	isEmpty
新增	push	enqueue
移除	pop	dequeue
擷取	getTop	getFront

密碼 cipher key



My Notes 12/2

Important Concepts worth keeping

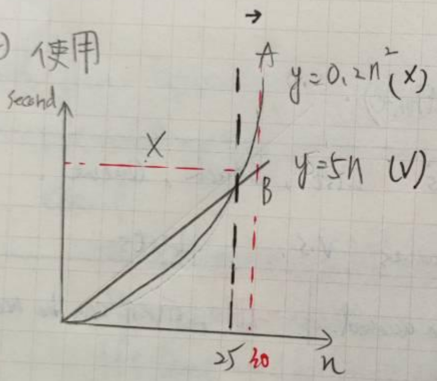
Today: / /

Big O Notation 判斷方法好壞(效率)

(1) 判斷兩種 { ① 時間效率
② 空間效率

※ 比較「顯著差異」

(2) 使用



隨 n 增加
A: 指數成長
B: 線性成長

比較 n=30 A、B 所花時間: A>B,
所以 B 為較有效率的方法

Big O Notation 寫法:

A is $O(n^2)$ - order n^2 (x)

B is $O(n)$ - order n (v)

My Questions
Problems & Difficulties needing exploration

My learning weather report
□ ☀
□ ☁
□ 🌧
□ ⛈

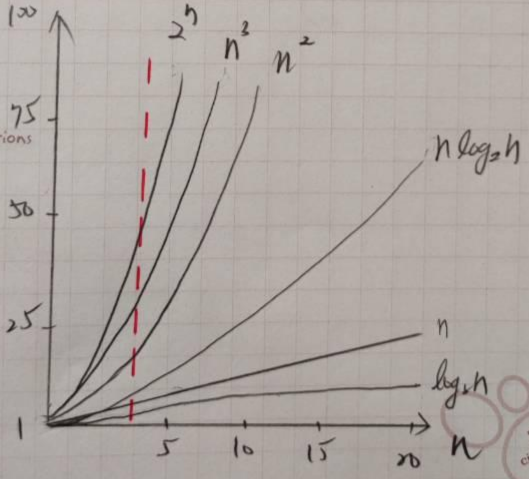
(3) 常見

效率 高 ↑ … ↓ 低

$O(1) \to n^0$	constant time	常数
$O(\log_2 n)$	logarithmic time	对数
$O(n)$	linear time	线性
$O(n\log_2 n)$		
$O(n^2)$	quadratic time	平方
$O(n^3)$	cubic time	立方
$O(2^n)$	exponential time	指数

My Opinions
Thoughts, inspirations, and suggestions

value of growth-rate function



密碼 cipher key