1. sort
2. binary search
3. two pointer
4. bfs dfs (queue, stack)
5. dynamic programming
6. bit operation
7. stack
8. queue
9. priority\_queue

演算法複習

KMP, Boyer-Moore, Quick\_select

做題目時 須確定 for 迴圈結束後 **“需不需要再判斷一次”**

如果是做進位的題目, 記得有可能最後carry = 1

1. sorting 可能發生地方
2. 組成一個一個pair 或group 來達成目的
3. 題目有提到想要unique不重複的解 或是有限個數unique的解
4. stack 使用地方
5. 當序列有遞增 或遞減 或有高有低 可能需要stack來解
6. 題目有要求 求下一個 “next …bahbah” 可能需要用stack
7. 逆序 reverse order. Ex. reverse linkedlist, reverse string
8. XOR problem
9. 題目提到找到多餘的 **“1”** 個
10. Map or set 使用地方 instead of using unordered\_map or unordered\_set

Ex. Absolute k diff, or need to sort

1. 區間題目解題方式
2. 區間overlapping 則合併區間
3. 區間non overlapping
4. Find median 技巧
5. 利用 two priority\_queue
6. 利用 quick select
7. nth\_elelment API
8. Circular Search 技巧
9. Priority\_queue 使用時機
10. Top K bahbah
11. Resovoir sampling
12. Tree 解題技巧
13. Queue
14. Stack …

Others coding attention !!!

1. <<= shift left, >>= shift right
2. 當 ~1 means 111111111111 31個1 後面為0 =>> -2
3. 當使用 迴圈做判斷時, 須想想 迴圈結束後 需不需要在再判斷一次

* Reshape the Matrix

(1) origin\_r \* origin\_c 的矩阵 reshape为 r\*c的矩阵，需要满足：   
origin\_r \* origin\_c=r\*c

(2)元素位置对应的关系   
如果将矩阵横向展开为一维数组，元素个数为n=origin\_r \* origin\_c   
在元素在一维数组中对应的位置i：   
原矩阵位置[i/origin\_c,i%origin\_c]   
新矩阵位置[i/c,i%c]

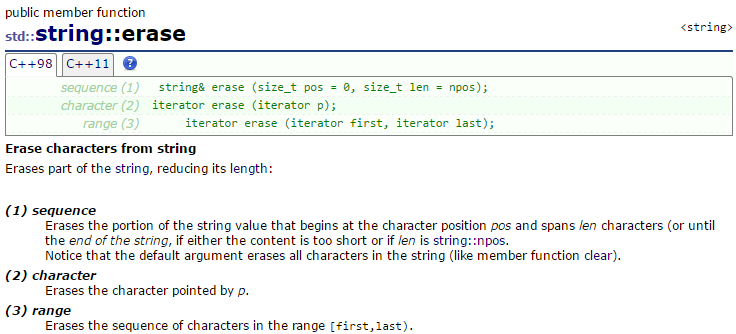
C++ string API

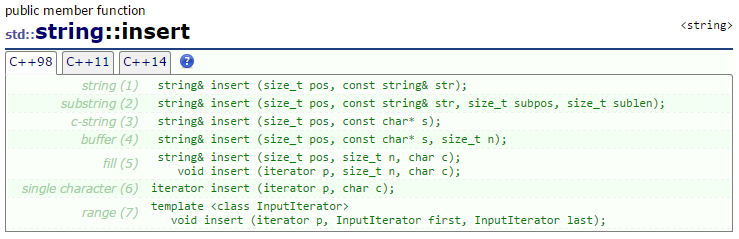
Need Attention

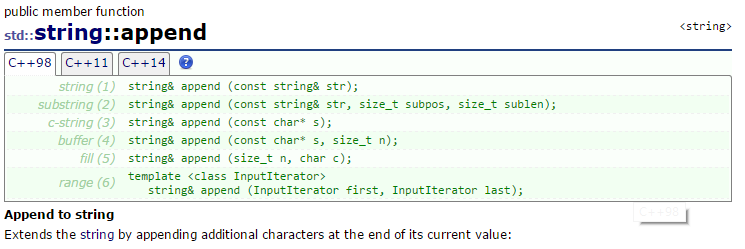
* Subsequence Problem

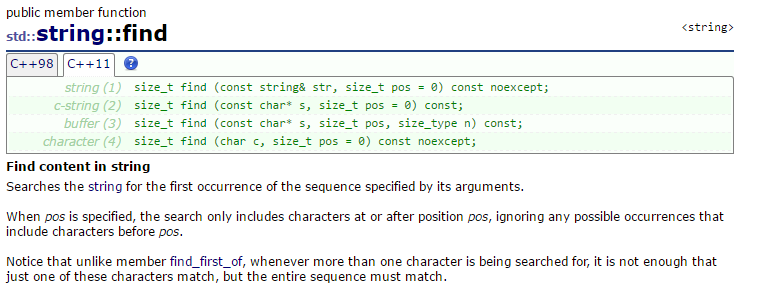
1. 如果只是判斷是不是subsequence?

A. 用while 迴圈and two pointers判斷 B. 用 dynamic programming









"-1/2+1/2+1/3"

stringstream ss(expression);

int a1, b1;

int a2, b2;

char delimeter;

char op;

ss >> a1 >> delimeter >> b1;

while(ss >> op >> a2 >> delimeter >> b2) {

}