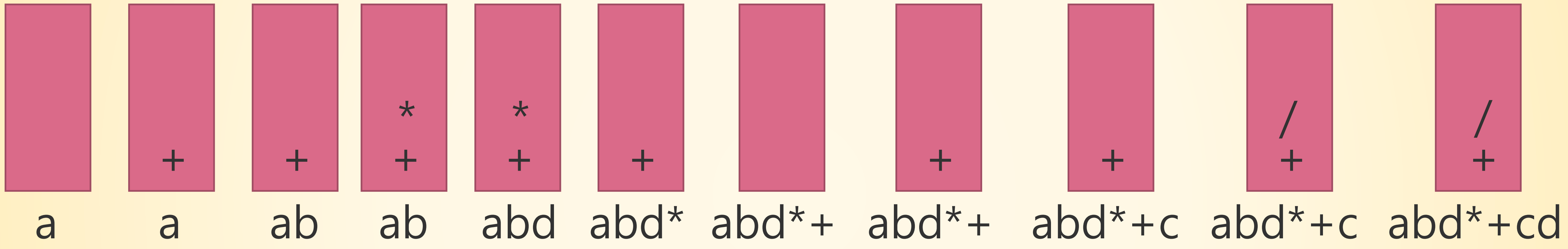
Ch. 3 Stack & Queue 參考答案

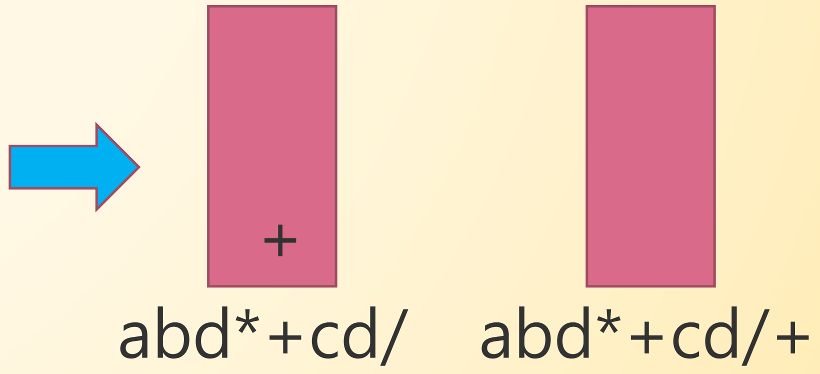
Question 1

Please use stack to convert the following infix notation to postfix notation and show the state of stack step-by-step. Please also describe your algorithm.

$$a+b*d+c/d$$

(Question 1) Ans: abd*+cd/+





參考: https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/

Question 2

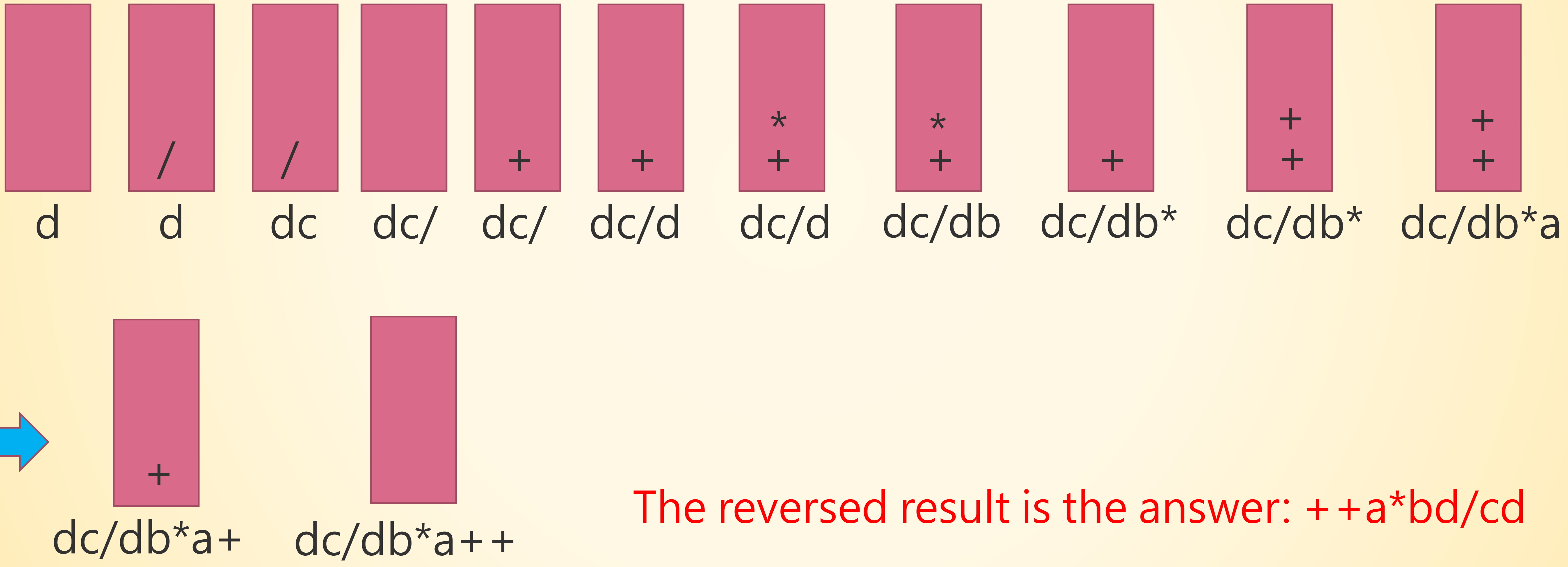
Please use stacks to convert the following infix notation to prefix notation and show the state of stack step-by-step. Please also describe your algorithm.

a+b*d+c/d

(Question 2) Ans:

Ans: ++a*bd/cd

(Note: Scan infix from right to left and apply the same algo.)



The reversed result is the answer: ++a*bd/cd

Question 3

1. Convert the following infix notation to pretfix /postfix.
2. An easy way to evaluate an infix expression for a computer is:
   a) Convert the infix expression to postfix.
   b) Evaluate the postfix expression by scanning from left to right.

   Please evaluate the following infix expression by first converting it to a postfix expression. (put in a=8,b=4,c=5,d=6,e=7)

a/b-c+d*e-a*c

(Question 3) Ans:

Prefix: -+-/abc*de*ac

Postfix: ab/c-de*+ac*-

Evaluation = -1

Evaluation 參考: https://www.geeksforgeeks.org/stack-set-4-evaluation-postfix-expression/

Question 4

Convert the following infix notation to pretfix/postfix notation and explain your process.

(a/(c*(b+d)))/(e-a)*c

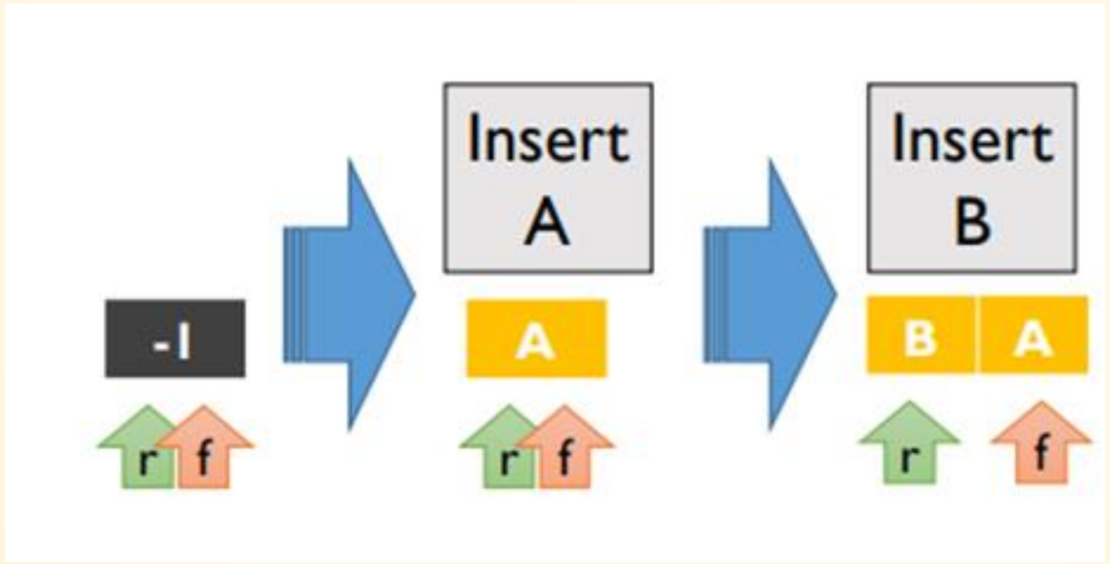
(Question 4) Ans:

Prefix: */ /a*c+bd-eac

Postfix: acbd+*/ea-/c*

Question 5

What's is the issue if a queue has finite capacity? Consider the following case if the capacity is only two.



(Question 5) Ans:

When the capacity is almost full, it becomes tricky for programmer to decide how to expand the capacity.

Question 6

The characteristic of the queue is first in first out. There is a way to implement a queue using two stacks. Suppose that you have one queue “A” and two stacks “B” and ”C”. Please use the push and pop functions of the stack to implement the push and pop functions of a queue.

(Question 6) Ans:

- 參考: https://www.geeksforgeeks.org/queue-using-stacks/

Question 7

The characteristic of the stack is last in first out. There is a way to implement a stack using two queues. Suppose that you have one stack “A” and two queues “B” and ”C”. Please use the push and pop functions of queue to implement the push and pop functions of a queue.

(Question 7) Ans:

- 參考: https://www.geeksforgeeks.org/implement-stack-using-queue/

Question 8

Describe what is Deque (double ended queue) and how it works?

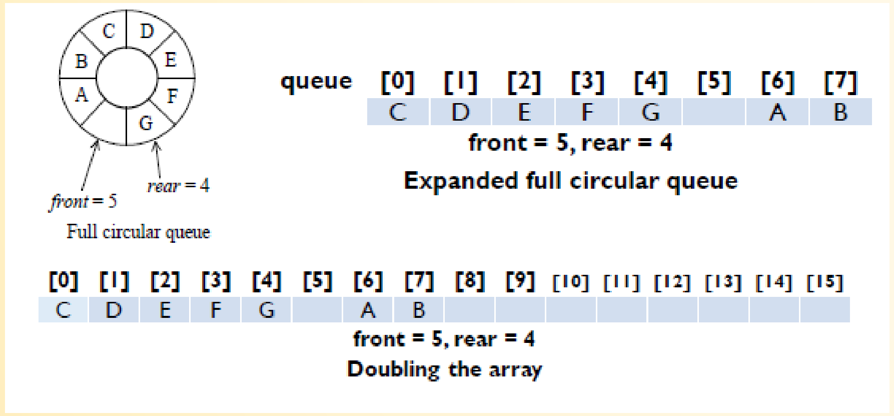
(Question 8) Ans:

A double-ended queue (deque) is an abstract data type whose elements can be added to or removed from either the front or back.

push_rear(): push into rear.
push_front(): push into front.
pop_rear(): pop from rear.
pop_front(): pop from front.
raer(): return the element at the rear.
front(): return the element at the front.

Question 9

According to the lecture video, we know that when a circular queue reaches its full capacity, we will double the queue capacity. However, it will cause some misplaced elements in the new circular queue. For example, the A and B in the following example are misplaced after doubling queue capacity. Please discuss and compare the pros and cons of the two scenarios.



Full circular queue

queue	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	C	D	E	F	G		A	B

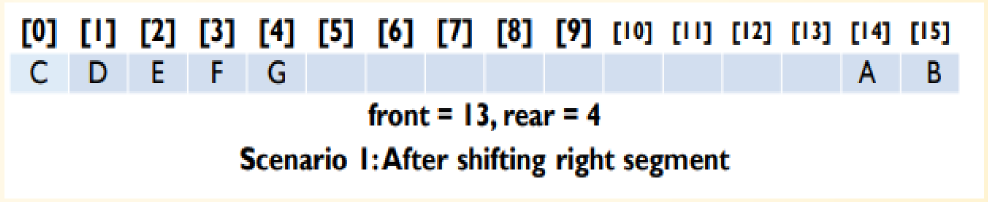
front = 5, rear = 4

Expanded full circular queue

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
C	D	E	F	G		A	B								

front = 5, rear = 4

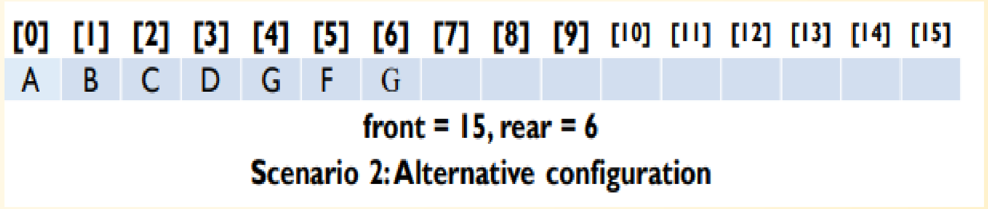
Doubling the array



[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
C	D	E	F	G										A	B

front = 13, rear = 4

Scenario 1: After shifting right segment



[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
A	B	C	D	G	F	G									

front = 15, rear = 6

Scenario 2: Alternative configuration

Ans: Question 9

1. How many elements that need to be moved?
2. How hard to write the code?

Question 10

According to the lecture video, when the capacity of a stack is insufficient to take new items, one needs to execute ChangeSize1D(stack, capacity, 2*capacity). Please give a description of how to implement ChangeSize1D.

Ans: Question 10

```
ChangeSize1D(s, oldSize, newSize){
        count = min(oldSize, newSize);
        copy “count” elements from s.storingArray to a temporary array;
        delete the original s.storingArray and replaced it with temp;
}
```

Question 11

Let b_n be the number of different permutations obtainable by passing the numbers 1, 2, 3, ..., n through a stack and doing push and pop all possible combinations.

(a) Give a recursive formula for c_n.
(b) Derive an analytical formula for c_n

For example, assume that we have 1 2 3 three elements, then there are 5 possible push-pop combinations. Here we list three of them for your reference.

1) push1, push2, push3, pop3, pop2, pop1. So the order is 321

2) push1, pop1, push2, pop2, push3, pop3. So the order is 123

3) push1, push2, pop2, pop1, push3, pop3. So the order is 213

(Question 11) Ans:

Scan from left to right, total encountered pop() at any moment should never be larger than push().

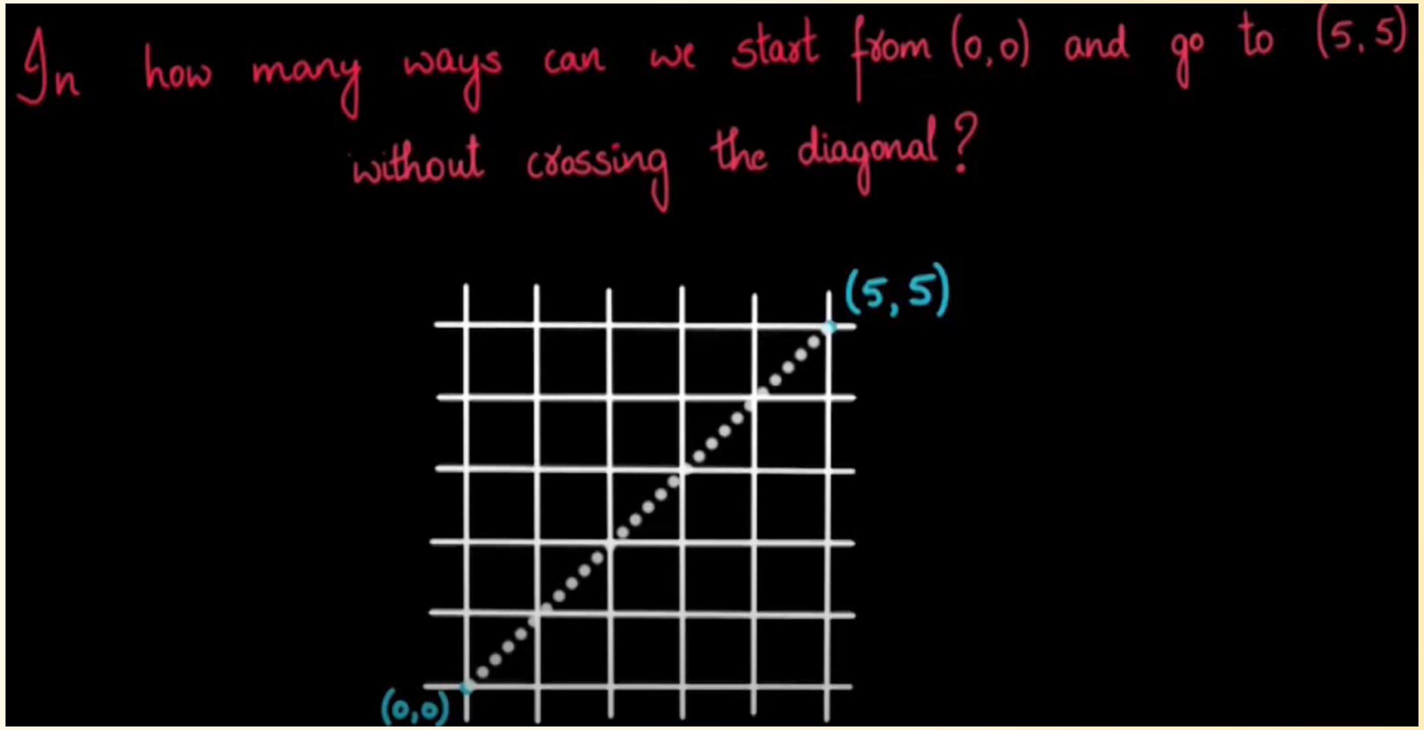
"Catalan Number": C_n

(a) $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}\, , C_0 = 1$

(b) $C_n = \binom{2n}{n} - \binom{2n}{n+1} = \frac{1}{n+1}\binom{2n}{n}$

(Question 11) Why $C_n = \binom{2n}{n} - \binom{2n}{n+1}$?

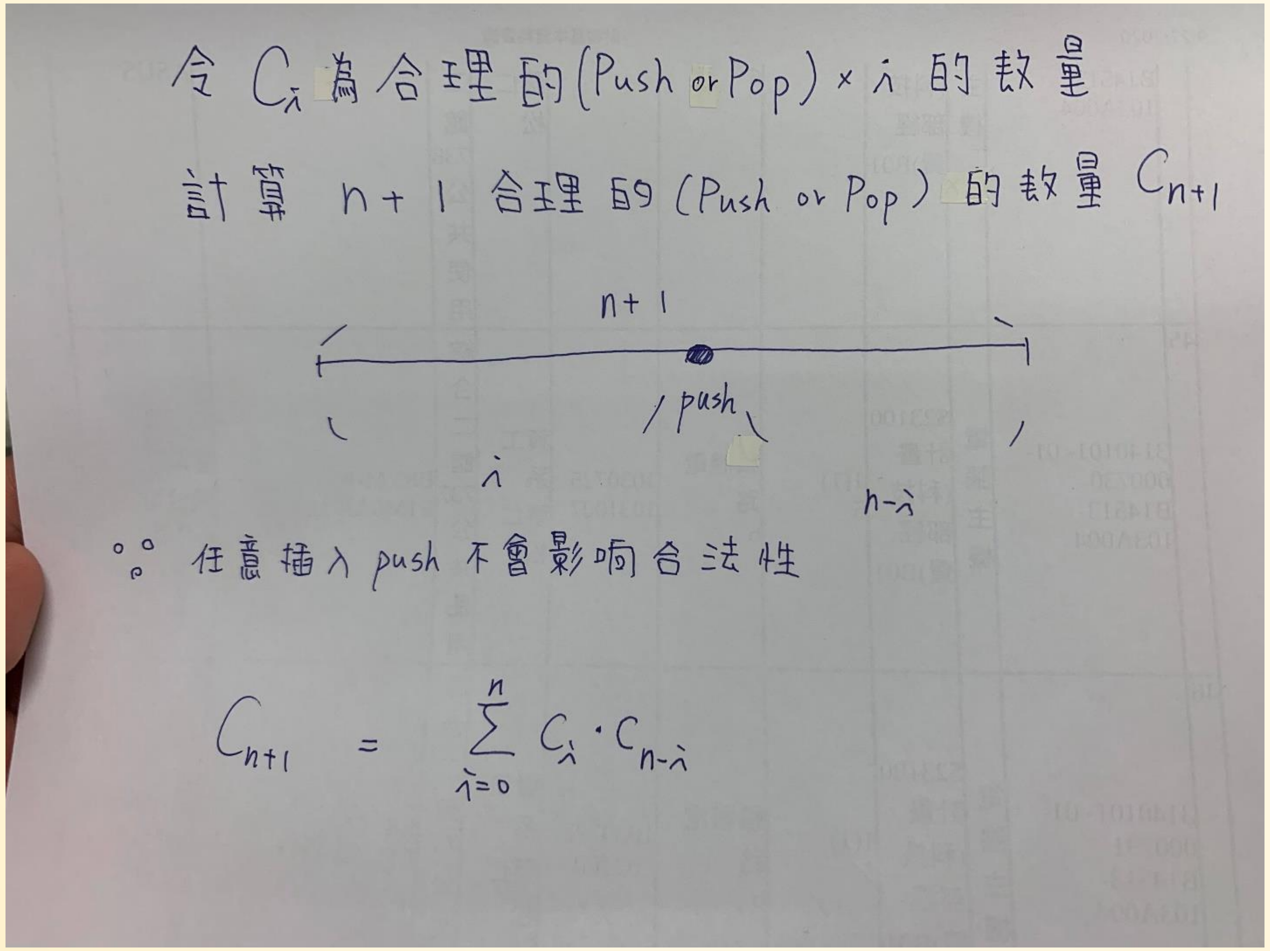
參考: https://math.stackexchange.com/questions/2633831/number-of-paths-in-a-grid-below-a-diagonal


In how many ways can we start from (0,0) and go to (5,5) without crossing the diagonal?
(5,5)
(0,0)

- R: go right one step, U: go up one step
- Total possible paths: 5U and 5R $=\binom{10}{5}$
- Valid: RRRUURURUU
- Invalid: RUUUURURRR or RURRUUURRU

URRUURURRR **URUURRR**RRU: both strings have 4U and 6R. There is a one-one mapping between these strings and the invalid strings. So the number of invalid strings are $\binom{10}{4}$ or $\binom{10}{6}$.

(Question 11) The Idea of $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$


令 C_i 為合理的 (Push or Pop) × i 的數量

計算 n+1 合理的 (Push or Pop) 的數量 C_{n+1}

∵ 任意插入 push 不會影響合法性

$$C_{n+1} = \sum_{i=0}^{n} C_i \cdot C_{n-i}$$