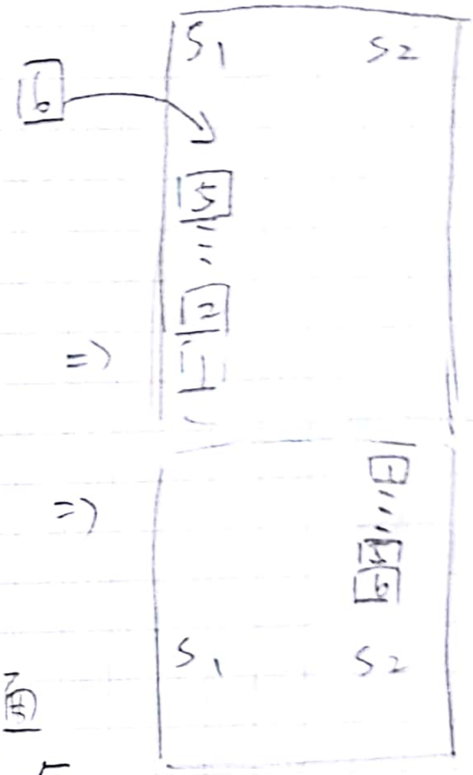
1. (1) Stack is LIFO

用 stack 來作出 enqueue

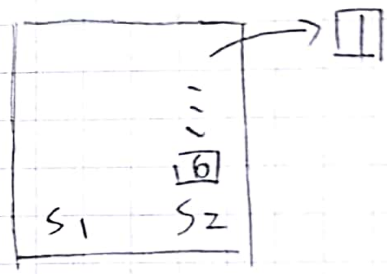
令 stack1, stack 為 s_1, s_2.

enqueue $\begin{cases} S1.\text{push}(1, 2, 3 \dots) & \Rightarrow \\ S_2.\text{push}(s1.\text{pop}()) & \Rightarrow \end{cases}$



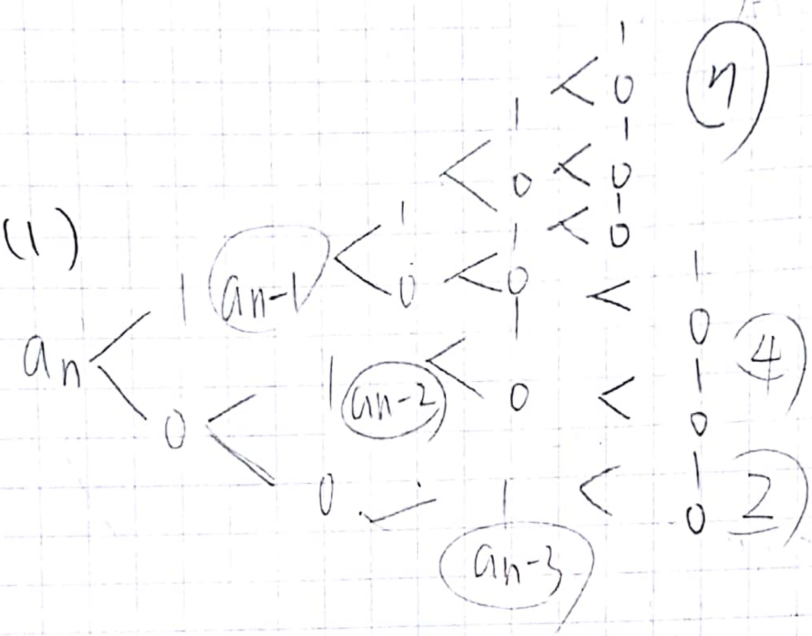
queue 之性質為 FIFO，舊的在前面
故符合

(2) S_2.pop() =>



將 stack2 中之值從高至低 pop 出去。

2、 (1)



$a_n = a_{n-1} + a_{n-2} + a_{n-3}$

$\quad\quad 7 \quad\quad 4 \quad\quad 2$

(2) 由(1)可看出，$a_1 = 2$，$a_2 = 4$，$a_3 = 7$

(3)

a_1 2

a_2 4

a_3 7

a_4 13

a_5 24

a_6 44

a_7 81

a_8 149

a_9 274

a_{10} 504 #

3. pop和push之量需要相等 balance

排列出之組合有 $C^{pop,push}_{(n,n)}$ 種，但因有stack之限制，不會每種路線均可行。此問題可以轉化為走格子問題，而3,1,2，並不少考量入此轉化問題。

when element 1 $C_1 = 1$

element 1,2 $C_2 = 2$

element 1,2,3 $C_3 = 5$

element 1,2,3,4 $C_4 = 14$

符合 Calatan Numbers

可以得出 $C_n = C^{2n}_{n} - C^{2n}_{n-1}$ ← 不合法之方式

$$= \frac{(2n)!}{n!\,n!} - \frac{(2n)!}{(n-1)!\,(n+1)!}$$

$$= \frac{(2n)!}{n!\,(n-1)!}\left(\frac{1}{n} - \frac{1}{n+1}\right)$$

$$= \frac{(2n)!}{n!\,(n-1)!} \times \frac{1}{n(n+1)}$$

$$= \frac{1}{n+1} C^{2n}_{n}$$

#

4、

寫出 Fibonacci

0 1 1 2 3 5 8 13 21 ...

採用數學歸納法

$F_1 \times F_3 - (F_1)^2 = -1$ 成立

$0 \quad 1 \quad 1^2$

$F_2 \times F_4 - (F_3)^2 = (-1)^2$ 成立

$2 \times 1 - 1^2$

若在 $n \leq k$ 時亦成立

則 $n = k+1$ 時：

$F_k \times F_{k+2} - (F_{k+1})^2$

$\Rightarrow F_k(F_k + F_{k+1}) - (F_k + F_{k-1})^2$

$\Rightarrow F_k^2 + F_k F_{k+1} - F_k^2 - 2F_k F_{k-1} - F_{k-1}^2$

$\Rightarrow [F_k^2 + F_{k+1} F_k - F_{k-1}(2F_k + F_{k-1})] - F_k^2$

$\Rightarrow [F_k^2 + F_{k+1} F_k - F_{k-1}(F_k + F_{k+1})] - F_k^2$

$\Rightarrow [F_k^2 + F_{k+1} F_k - F_{k-1} F_k - F_{k-1} F_{k+1}] - F_k^2$

$\Rightarrow F_k(F_k + F_{k+1} - F_{k-1}) - F_{k-1} F_{k+1} - F_k^2$

$\Rightarrow F_k(F_k + F_k + F_{k-1} - F_{k-1}) - F_{k-1} F_{k+1} - F_k^2$

$\Rightarrow 2F_k^2 - F_k^2 - F_{k-1} F_{k+1} = (-1)[F_{k-1} F_{k+1} - F_k^2]$ ##