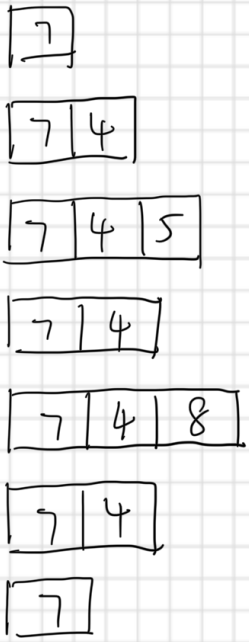
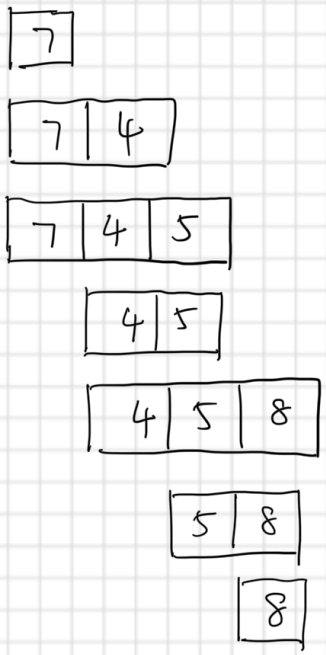
DSAA 8　　12310401　王子恒.

8 1. 1.



2.



3. L. hand → | 1 | 7 |

L. hand → | 1 | 4 | ⇄ | 7 |

L. hand → | 1 | 5 | ⇄ | 4 | ⇄ | 7 |

L. hand → | 1 | 5 | ⇄ | 7 |

L. hand → | 1 | 8 | ⇄ | 5 | ⇄ | 7 |

L. hand → | 1 | 8 | ⇄ | 5 | 1 |

L. hand → | 1 | 5 | 1 |

8.2 S_1 can start from the beginning of the array and grow towards the end, while S_2 can start from the end of the array and grow towards the beginning.

```
topS1 = 0
topS2 = n+1

function PushS1(A, x)
    if topS1 < topS2 - 1
        topS1 = topS1 + 1
        A[topS1] = x
    else
        error : "stack overflow

function PushS2(A, x)
    if topS1 < topS2 - 1
        topS2 = topS2 - 1
        A[topS2] = x
    else
        error : "stack overflow

function PopS1(A)
    if topS1 >= 1
        x = A[topS1]
        topS1 = topS1 - 1
        return x
    else
        error : "stack underflow"

function PopS2(A)
    if topS2 <= n
        x = A[topS2]
        topS2 = topS2 + 1
        return x
    else
        error : "stack underflow
```

8.3

```
front = 0
last  = 0
count = 0

function ENQUEUE(A, item)
    if count == maxsize:
        error "overflow"
    else:
        A[last] = item
        last = (last + 1) mod maxsize
        count = count + 1

function DEQUEUE(A).
    if count == 0:
        error "underflow"
    else:
        item = A[front]
        front = (front + 1) mod maxsize
        count = count - 1
        return item
```

8.4

```
stack S1
stack S2

function ENQUEUE(item)
    S1.push(item).

function DEQUEUE()
    if S2 is empty
        while S1 is not empty:
            x = S1.pop()
            S2.push(x)
    if S2 is not empty:
        x = S2.pop()
        return x
```

O(1).

best case: O(1)

worst case: O(n)

average: every element pop once from S1 to S2 and pop once from S2, thus the average runtime is 2 = O(1).