1. C

2. C

3. D

1. O(1) O(n)

O(n) O(n)

~~O(n²)~~ O(nlogn)

2. (4) insert->next = ~~new~~ node target->next

(5) target->next

Q1. LIFO

FIFO ✓ Undo 連式呼叫

緩衝 任務排隊

Q2 1. Avoid ~~memory leak~~ 防止 Dangling Pointer

2. 只需要改變指標的指向在刪之前

~~head 情況下~~ → 重連 target 前後節點即可

3. 需要 head 像 prev, next 額外

~~要 traversing 遍歷所有~~ Sequential Access

1. B

2. C

3. ~~A~~ B

4. ~~B~~

$O(n)$ $\Omega(1)$

$O(n^2)$ $\Omega(n^2)$

$O(1)$

$O(n)$

(5) prevTarget.next

(6) target->next

O(1) O(logn) O(n) O(nlogn)

newNode->next=head

head = newNode

head = head->next

insert->next = target->next

target->next=insert

push pop isem isfull

enqueue dequeue isem isfull

60 / 105