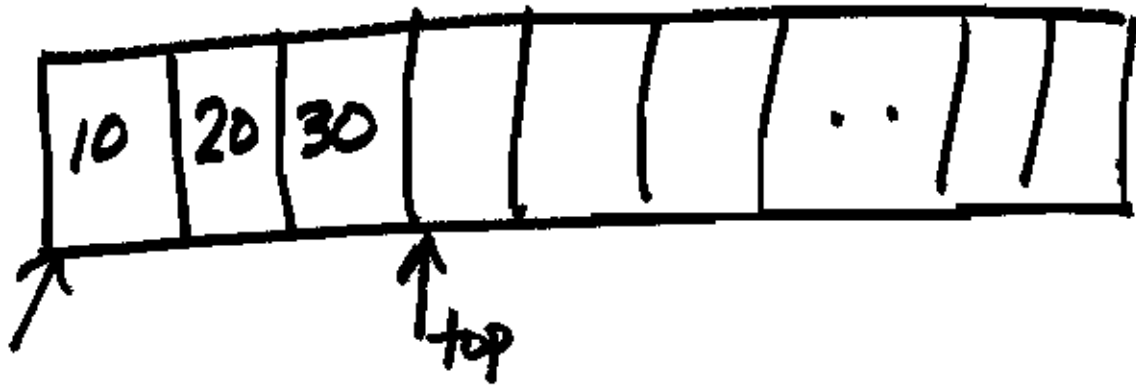


Client Interface

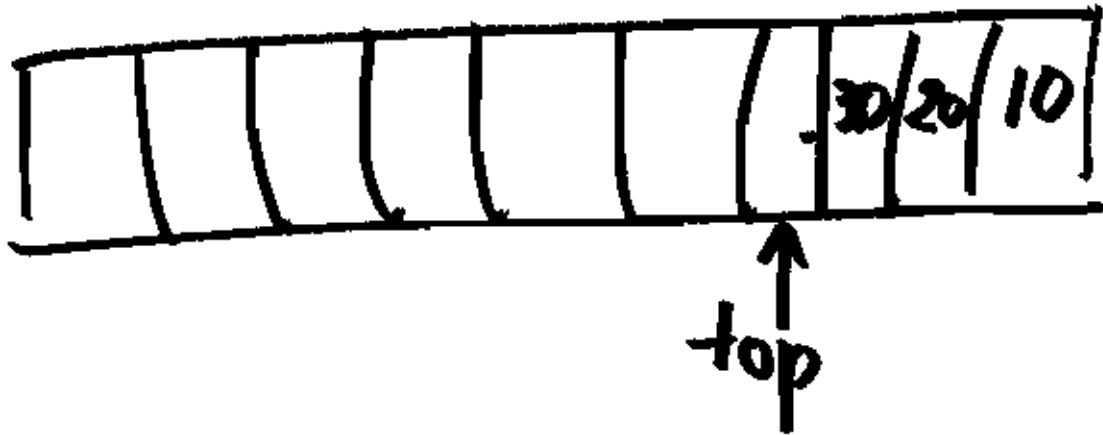
```
class stack {  
    public:  
        stack();  
        ~stack();  
        int push(const data & );  
        int pop(data &);  
        int peek(data &);  
        int isempty(); int isfull();  
        int pop();  
        int display-all();  
        stack (int);
```

Array - Stack

(1)



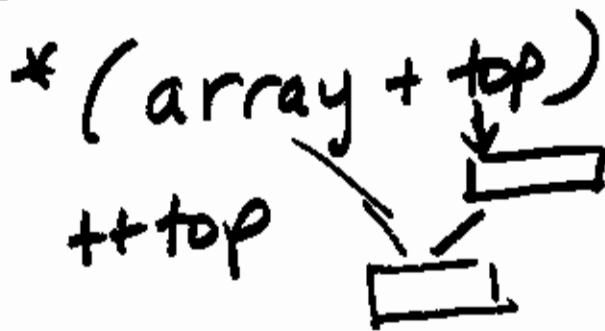
(2)



Run Time Performance

array[top] =

++top

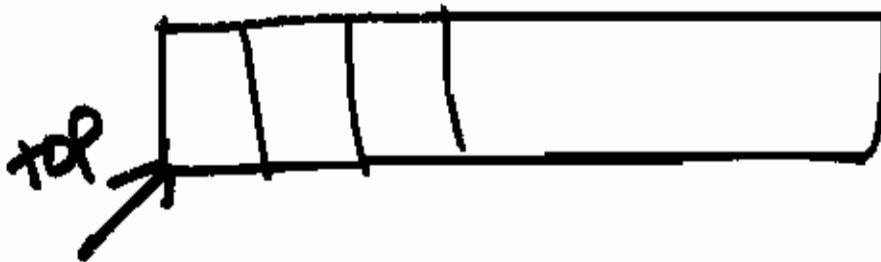


$\sim 4op + \sim 5\text{Add}$
 ~ 9

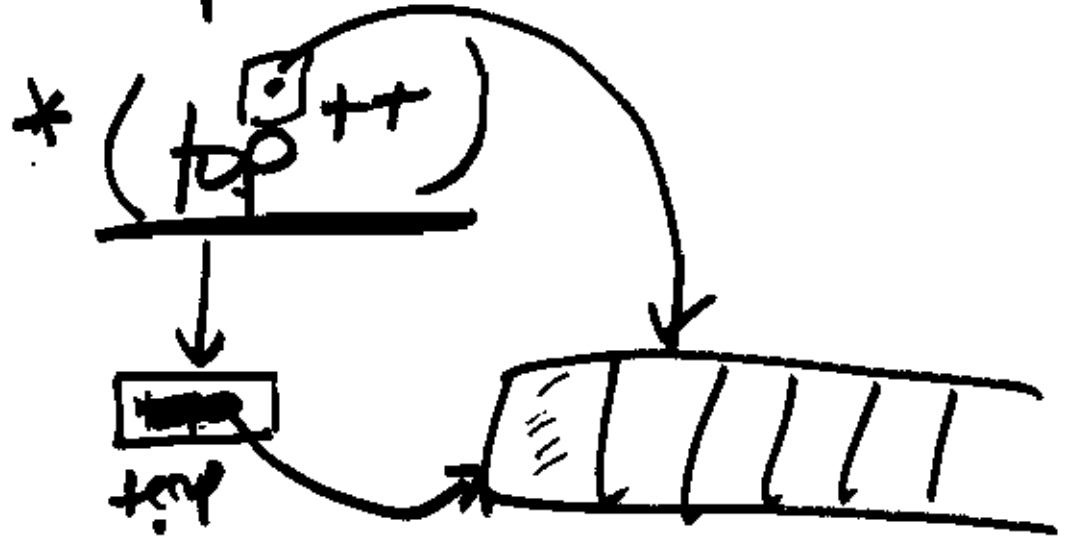
data * top = array; } top = &array[0]

*top++ =

$\&(*(\text{array} + 0))$
 $\& \text{array}$

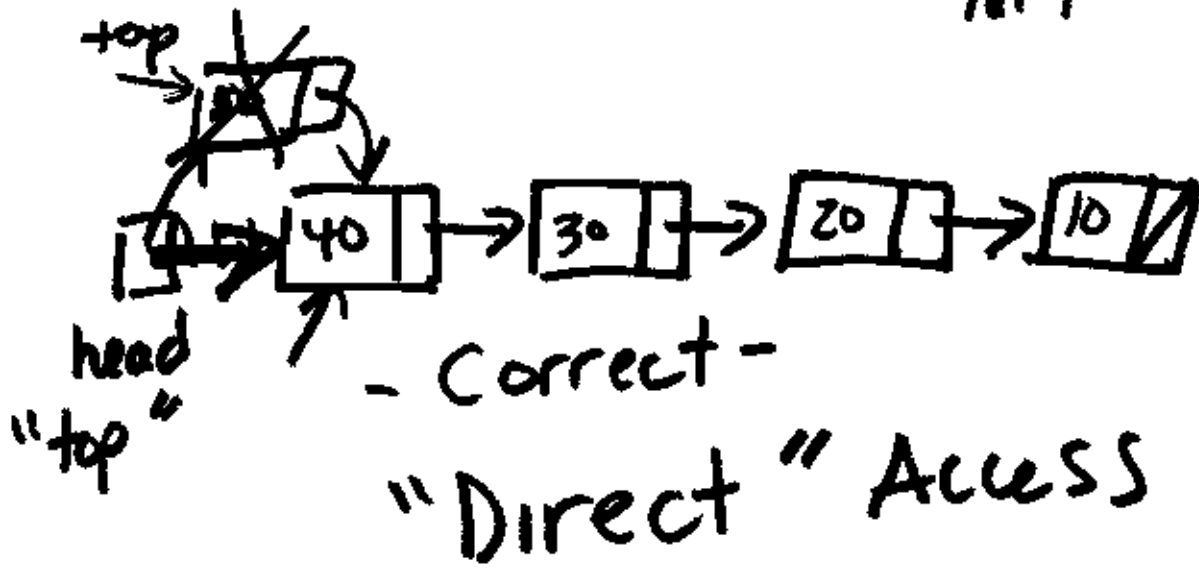
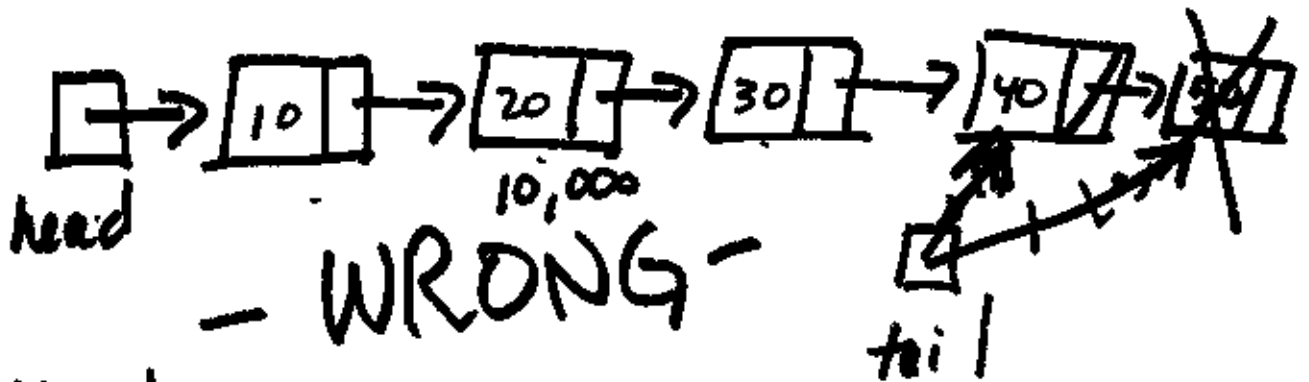


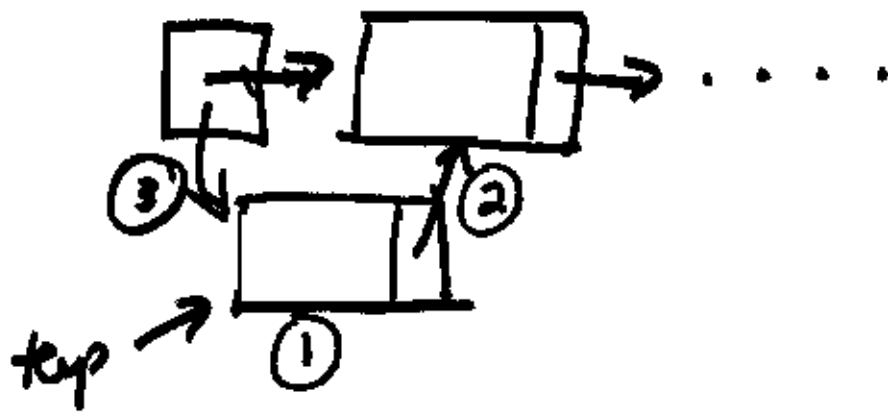
$*top++ = \dots$



$\sim 4op/fetches$

LLL - Stacks





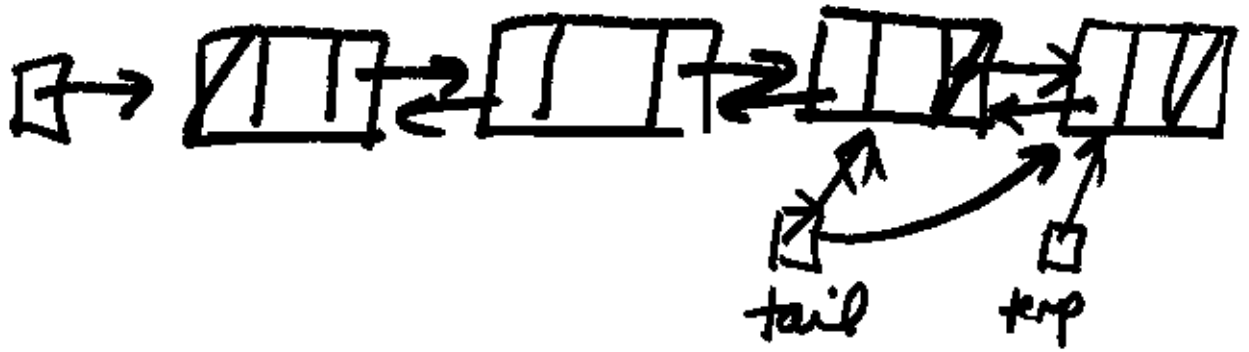
`temp = new node;`

`temp → data := ... // copy data`

`temp → next = head;`

`head = temp;`

$\sim 8 \text{ ops} + 4 = 12$



temp = new node;

temp → next = NULL

temp → data =

temp → prev = tail

tail → next = temp

tail = temp

13q + 7 = 20 ^{DLL}

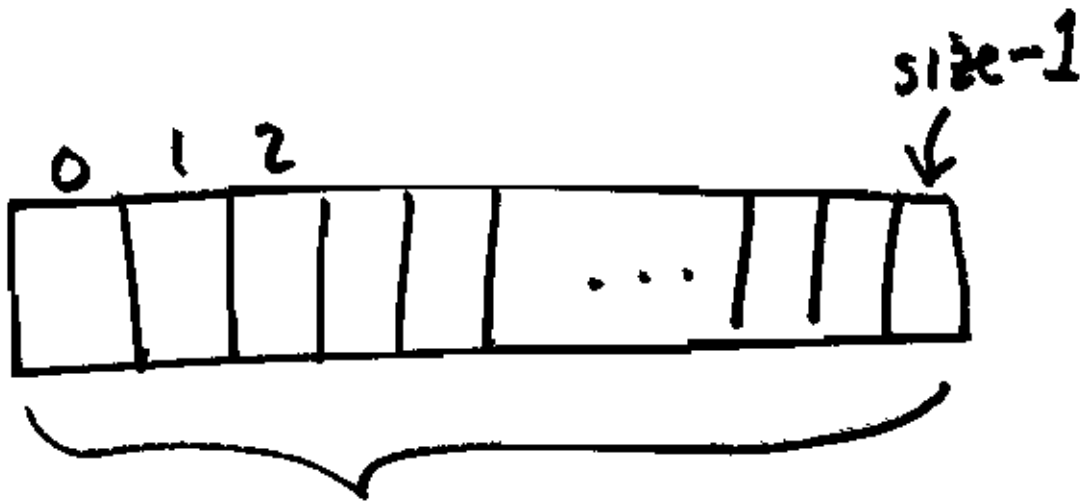
tail → next = new node;
tail → next → prev = tail
tail = tail → next;
tail → next = NULL;
tail → data =

$$18 + 8 = 26$$

Client Interface

```
class queue {  
    public:  
        queue(); queue(int);  
        ~queue();  
        int enqueue(const data &);  
        int dequeue(data &);  
        int peek(data &);  
        int isempty(); int isfull();  
        int display-all();  
        int dequeue();
```

Circular Array - Queue



index \% size

values $\phi \rightarrow \text{size}-1$

~~rear [index]~~

array [rear] =

$\text{rear} = (\text{rear} + 1) \% \text{size}$

++rear;

$\text{rear} = \text{rear} \% \text{size}$


++rear;
 $\text{rear \%} = \text{size}$

$++rear \% = size;$
↑
first

↑
lowest

NO

$rear++ \% = size;$



The diagram shows a circular array represented by a rectangle. A pointer labeled 'first' is positioned at the bottom center of the rectangle. An arrow points from the text 'rear++' above to the top edge of the rectangle, indicating the current position of the rear pointer. The entire diagram is enclosed in an oval.