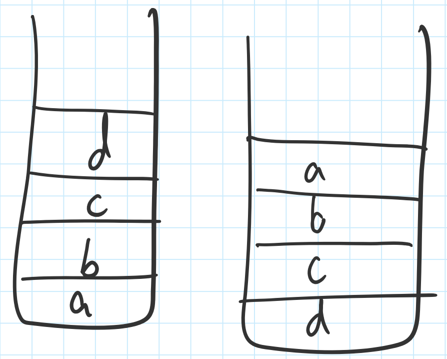
Stacks, Queues

31 March 2018 10:28



① usị 2 stks

② usị 1 stk.

<u>prev Smaller No</u>

4	5	2	6	8
✓-1	✓4	✓-1	✓2	✓2

<u>next Greater Int</u>

4	5	2	10	8
5✓	10✓	10✓	-1✓	-1✓

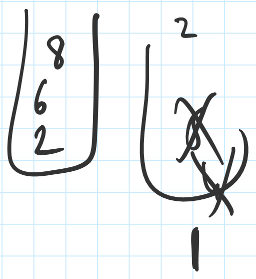
directory struct

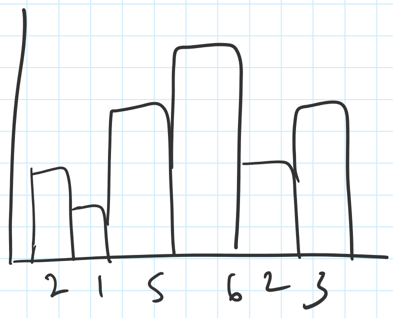
4 $\underbrace{2}$ 5 ?

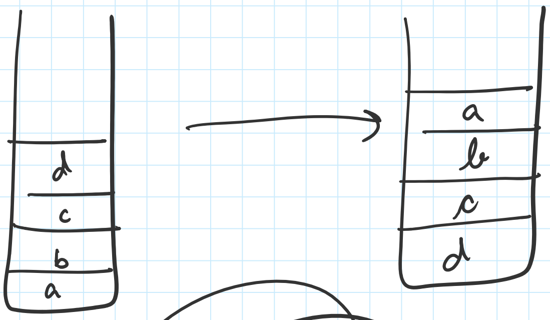
4 8 2 6 8

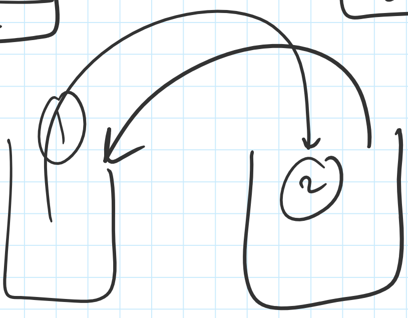
-1 4 -1 2 6

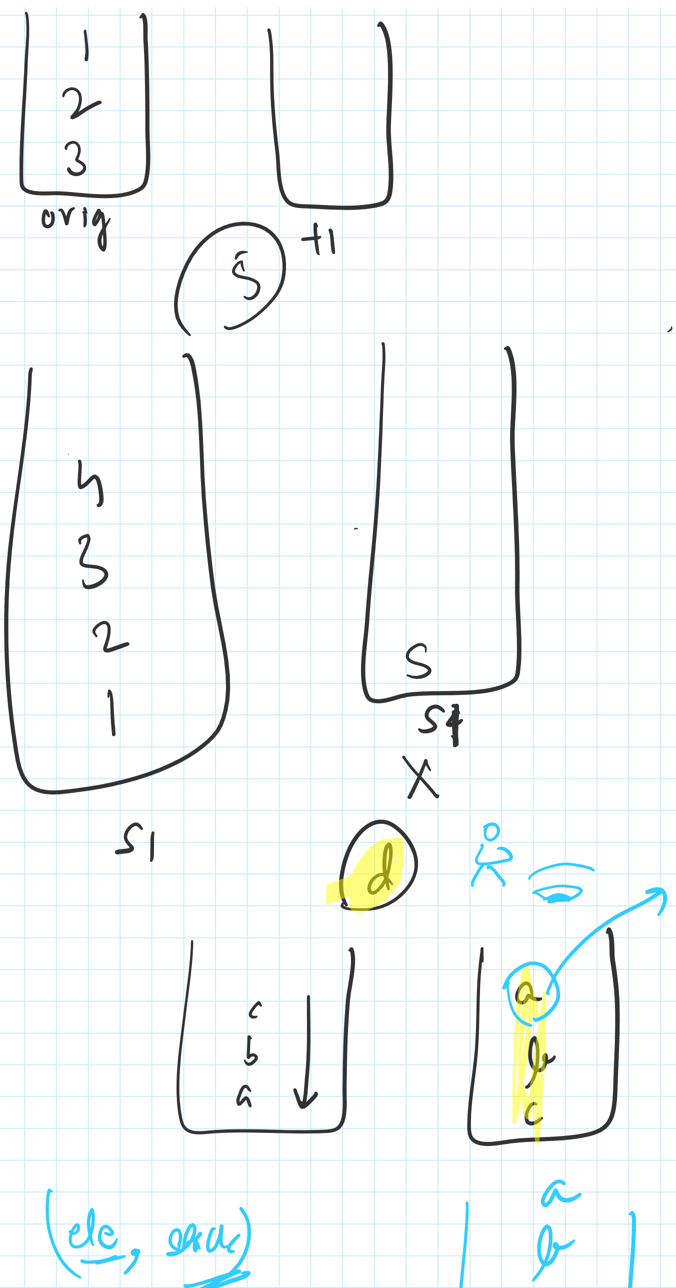
```
for i:[0,n) {
①  if (empty) {ans[i] = -1
              push(arr[i]);   |||
              }
   else {
✓    while (arr[i] <
              s.top() )  {   |
         s.pop(); }
            ①
   else ans[i] = s.top();
     s.push(arr[i]);   ||||
   }
```





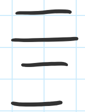






```
cnt = 0;
for ([1 to n-1]) {
    x = s1.top
    s1.pop();

    for (i = [1 to n-cnt-1])
        s1 -> s2;
        s1.pop();
    }
    ~~s2~~ [s1.push(x);
           ++cnt ]
    while (!empty) {
        s2 -> s1;
    }
}
```

```
template <typename T>
class Stack {
    T   B;

        push ( Book B)
        pop
    T top
}
```



```
Stack <Book> S;

Stack <plate> S;
```

e f a b c

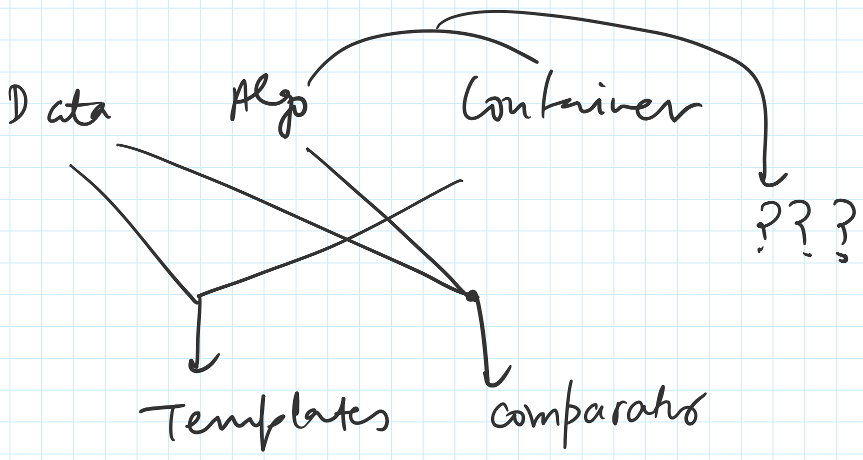
```
template <typename T>
           T      int
sort ( arr[], N,
     criteria)

tell( Book A , Book B) {
// shall A appear b4 B ??
```

```
template<typename
            T>
     T      int
sort( arr[], N )

if (arr[i] > arr[i+1])
```

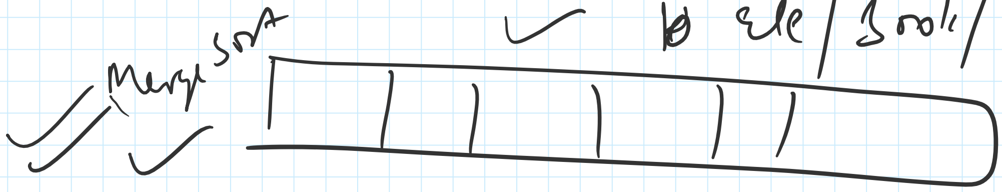
```
if (comp ( arr[i], arr[i+1]) {
     comparison
}
```

```
template <typename T>
void sort ( T arr [],
        int N,
        bool comp ( const T& a , const T& b)) {

    // own logic
}
```

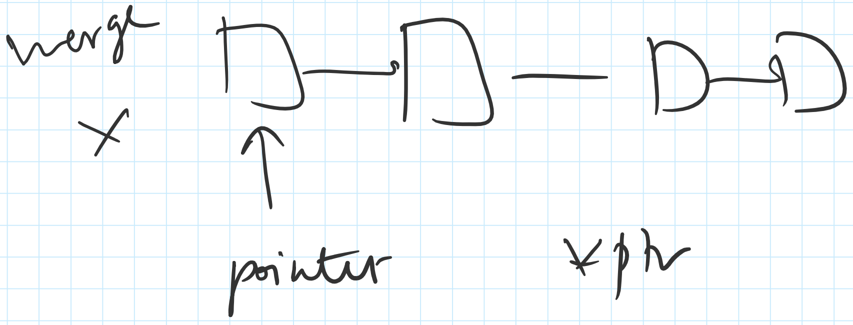
```
bool compmyEle ( Ele A , Ele B) {
     if (A.wt < B.wt) return ✓;
}            ret x;
sort < Elephants ( arr[ ], 20,
                   comparemyEle );
```



✓ ~~b~~ ele / book / students



<u>mid</u> = (i+j)/2;



<u>iterators</u>

Input	Output
Iter	Iter
(read)	(write)
(it++)	it++
(*it)	$\boxed{\text{*it = \_\_\_}}$

Forward it

*it

*it = ___

it++

↓

Bidirect it

--it;

↓

Random Access ; ++it; it += 5;