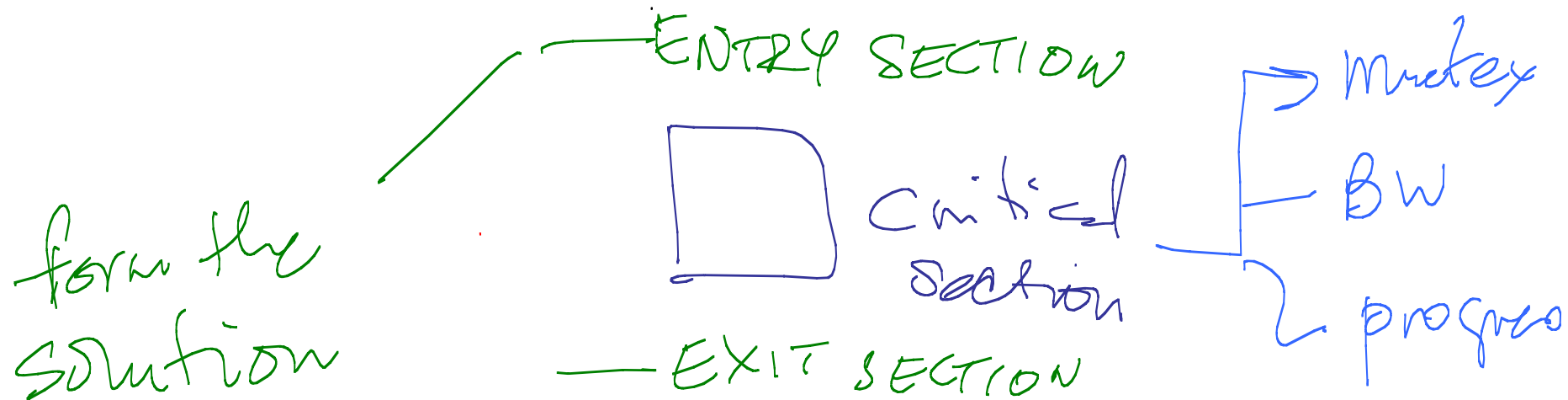


Find the correct 2-process  
software solution to the critical  
section problem



flag → 0, 1

Process 0

~~flag[0] ← 1~~  
while (flag[1] == 1) ;  
→ flag  
CS

flag[0] ← 0

flag[1] ← 1  
while  
flag[0] == 1 ;  
→ flag  
CS  
flag[1] ← 0

Process i

flag[i] ← 1

while (flag[j] == 1) ;

CS

flag[i] ← 0

Process 0

flag 0 & 1  
turn can be  
0 or 1

array  
integer

```
flag[0] ← 1;  
turn = 1  
while (flag[1] == 1) and  
(turn == 1);
```

BUSY  
WAIT

CS

```
flag[0] = 0;
```

Process 1

```
flag[1] ← 1
```

```
turn ← 0
```

```
while (flag[0] == 1)  
and  
(turn == 0);
```

CS

```
flag[1] = 0
```

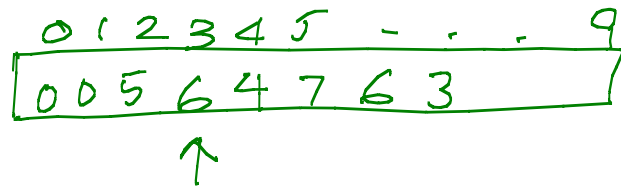
Lamparts Bakery

Algorithm

---

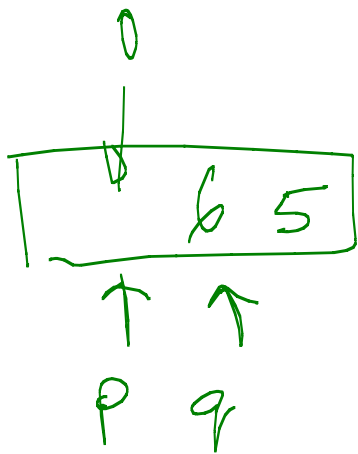
[ Choosing  $\rightarrow$  n-array  
num  $\rightarrow$  n-array ]

Process (i)



$num[i] = \max \text{ of } (num[0] \text{ to } num[n-1]) + 1$

for ( $j \rightarrow 0 \text{ to } n-1$ )



while ( $num[j] \neq 0$ ) and  
( $num[j], j < num[i], i$ )  
→ loop

CS

$num[i] = 0;$

Process(i)

choosing[i] = 1

num[i] = max + 1

choosing[i] = 0

for j = 0 to n-1 {

while (choosing[j] == 1);

while (num[j] ≠ 0) and

(num[j], j < num(i), i);

CS