자료구조 실습03

Data Structures Lab03

Lab03: Template를 이용한 Generic Class 구현

◎ 내용:

- ☞ Array를 기반으로 하는 Generic Queue class 정의 및 구현
- ☞ template을 이용한 Generic data type과 동적할당을 이용한 가변길이 Circular Queue Application 구현
- ☞ template을 이용한 Generic list class 구현
- ☞ Circular Queue의 응용

◎ 방법

- ☞ 배열을 기반으로 하는 Queue 동작원리를 이해하고 주어진 Stack의 소스코드를 참조하여 Circular Queue 구현
- ☞ Queue와 List의 특성을 가진 자료구조 설계 및 구현

◎ 제출물

- ☞ “Lab 03”에 정의된 응용 프로그램의 소스코드와 실행결과

예제: Template을 이용한 Queue 구현

◎ 목표: Template을 이용한 Queue class 설계 및 구현

◎ 내용:

- ☞ Array를 이용한 Generic Queue class 정의 및 구현
- ☞ 과제: template을 이용한 Generic data type과 동적 할당을 이용한 가변길이 Circular Queue Application 구현

◎ 방법

- ☞ 주어진 Stack 코드를 분석하여 가변길이 Circular Queue Application 구현
- ☞ 구현한 Queue를 template을 이용한 Generic class로 변환

Generic Circular Queue ADT

```
Template <typename T>
class CircularQueueType
{
public:
        CircularQueueType();                // Create circular queue(default size) using dynamic allocation
        CircularQueueType(int max);         // Create circular queue(max size) using dynamic allocation
        ~CircularQueueType();               // default destructor, release circular queue

        bool IsFull();                      // check the circular queue is full or not
        bool IsEmpty();                     // check the circular queue is empty or not
        void MakeEmpty();                   // make empty circular queue
        void EnQueue(T item);               // if circular queue has space then add item to top
        void DeQueue(T &item);              // if circular queue has any item then return item and delete it.
        void Print();                       // display all item on screen
private:
        int m_iFront;                       // front index of the circular queue
        int m_iRear;                        // rear index of the circular queue
        int m_nMaxQueue;                    // maximum size of the circular queue
        T *m_pItems;                        // item pointer
};
```

console

◎ Queue를 테스트할 driver는 다음과 같이 작성함

```
--- ID – Command -----
        1 : Enqueue Element
        2 : Dequeue Element
        3 : Is Empty?
        4 : Is Full?
        5 : EmptyQueue Exception test
        6 : FullQueue Exception test
        7 : Print all
        0 : Quit

Choose a Command -->
```

과제: Queue에 List 기능 추가

◎ 예제에서 구현한 Queue class에 다음과 같은 list class의 기능을 추가(새로 정의한 class를 “QList”로 명명)

- ☞ ResetList(): pointer가 front 가리키도록 세팅
- ☞ GetNextItem(): empty가 아니면 pointer를 증가시키고 pointe가 가리키는 item과 true를 리턴
- ☞ Search(item): item의 ID와 일치하는 레코드를 Queue에서 리턴
- ☞ Delete(item): item의 ID와 일치하는 레코드를 Queue에서 삭제
- ☞ Replace(item): item의 ID와 일치하는 레코드를 Queue에서 찾아서 item에 들어있는 정보로 대체

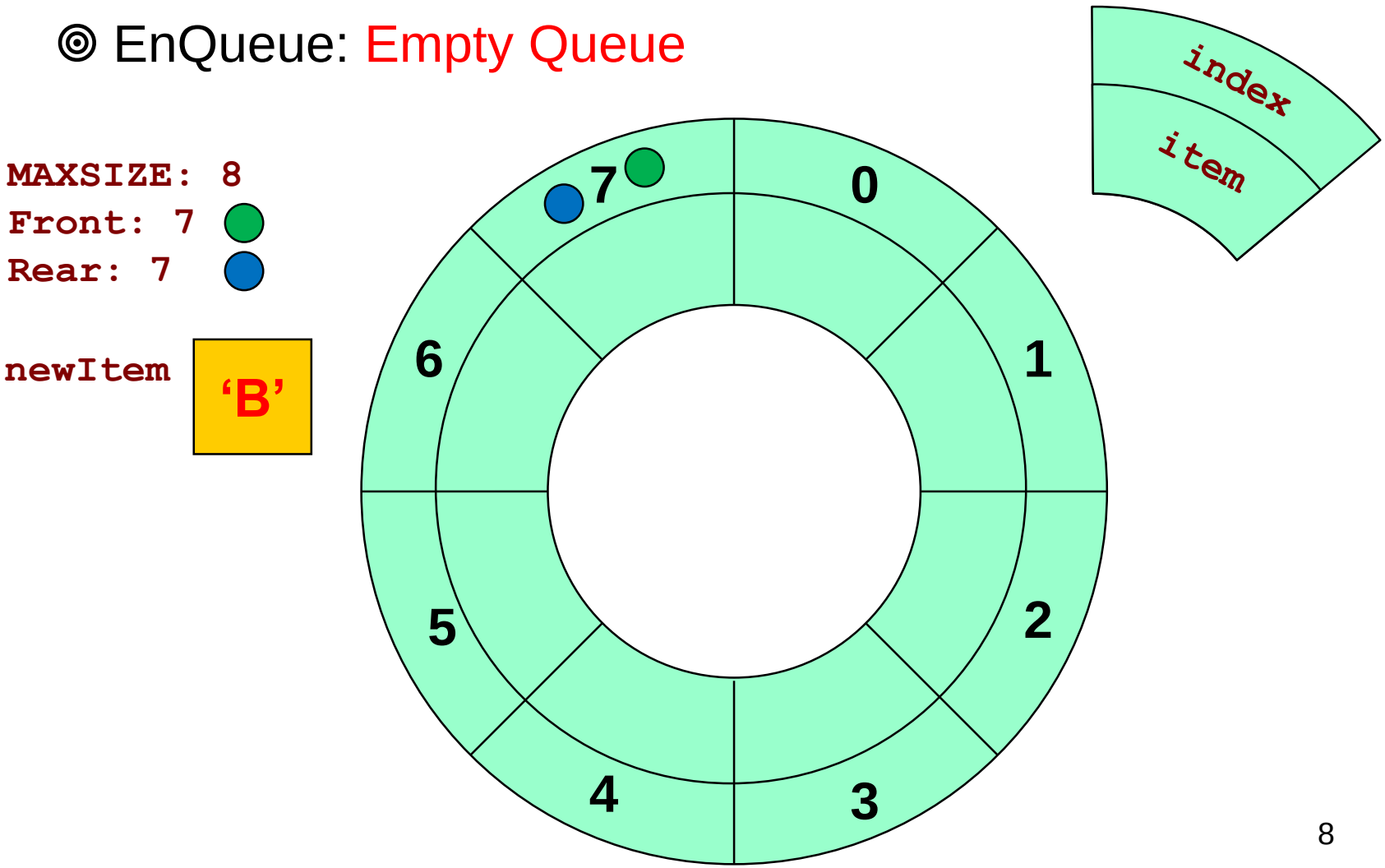
과제: Queue에 List 기능 추가

◎ QList의 기능을 테스트하기 위해서 application class 다음과 같은 기능을 추가한다.

- ☞ AddRecordToQL(): ID를 키보드로부터 읽어서 list class의 멤버함수를 이용하여 list에서 찾고 찾은 레코드를 QList에 Enque 추가
- ☞ DisplayQList(): QList에 저장된 항목을 화면에 출력
- ☞ DeleteQList(): 키보드로부터 ID를 읽어서 해당 레코드를 QList에서 찾아서 삭제
- ☞ Run(): 위에 추가된 기능을 테스트할 수 있도록 Run()함수를 수정

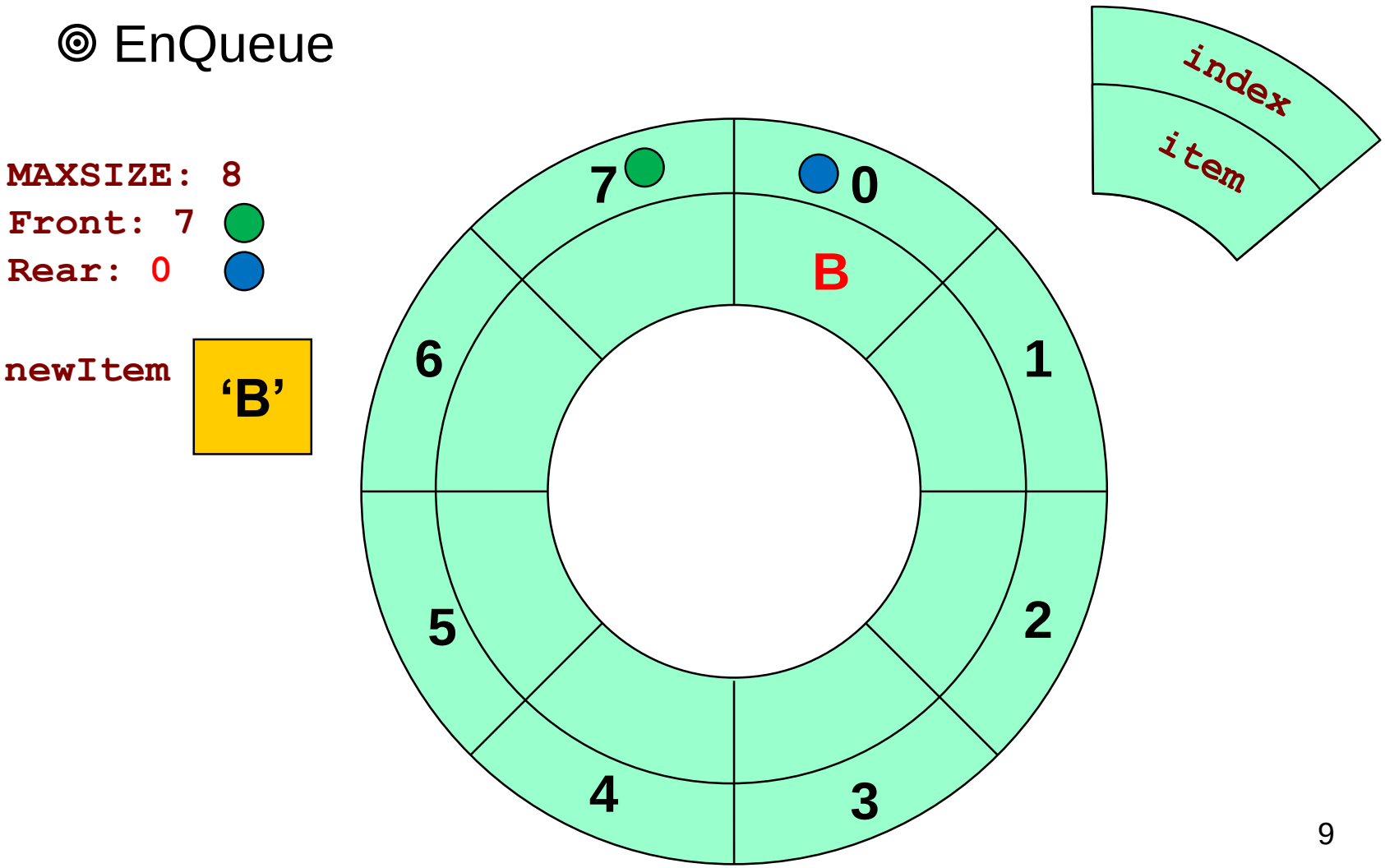
Lab03: Reference (1/3)

◎ EnQueue: Empty Queue



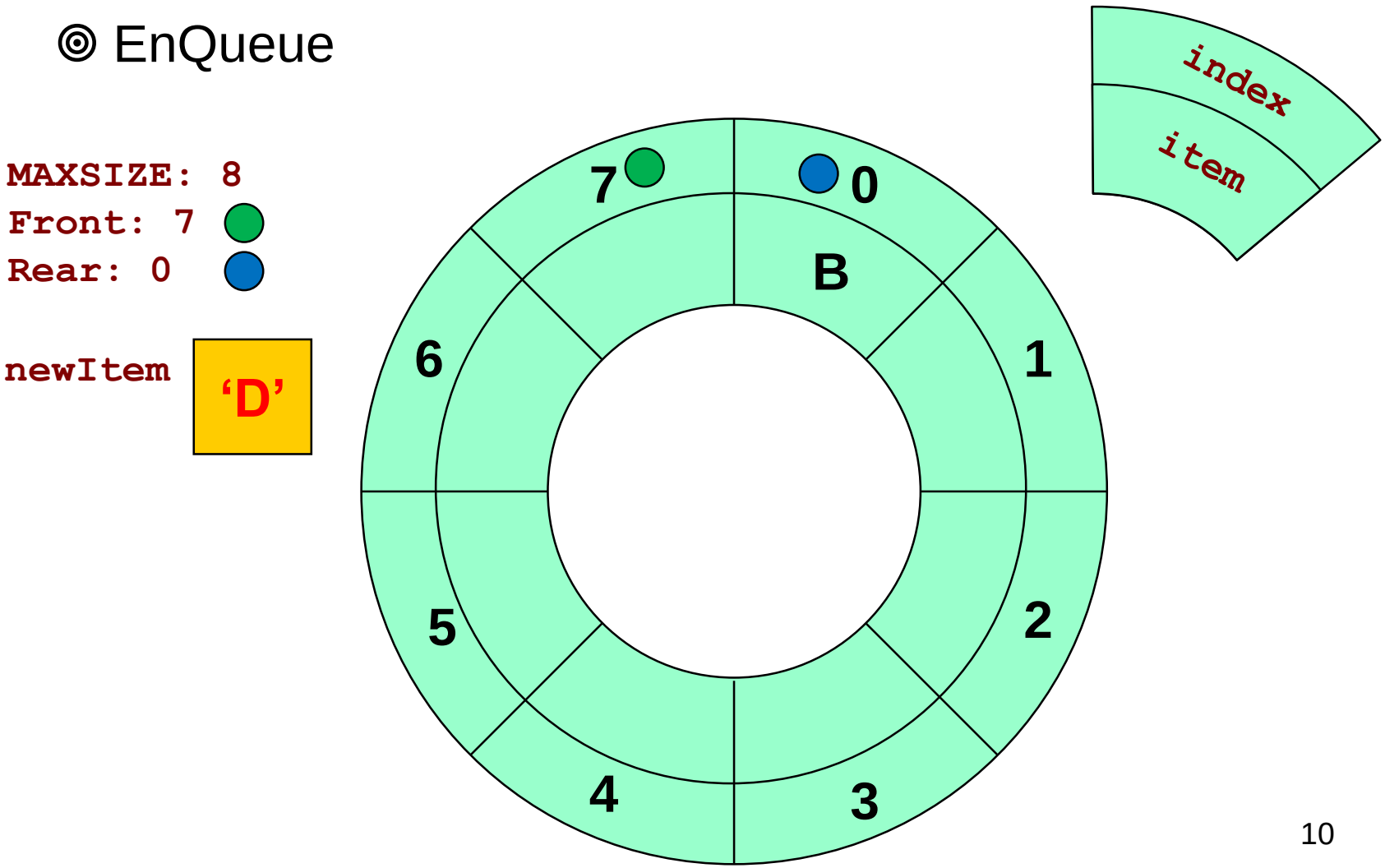
Lab03: Reference (1/3)

◎ EnQueue



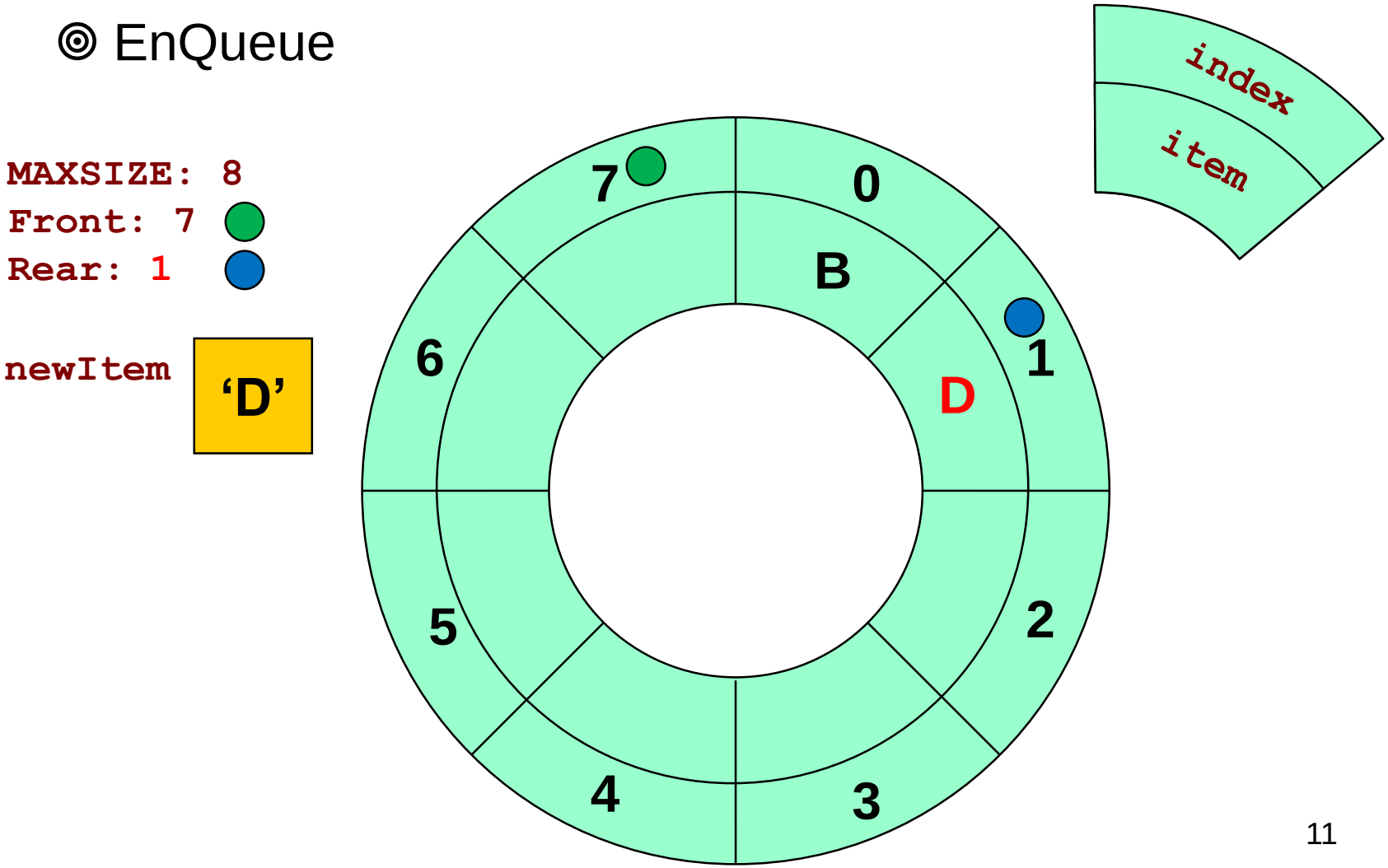
Lab03: Reference (1/3)

◎ EnQueue



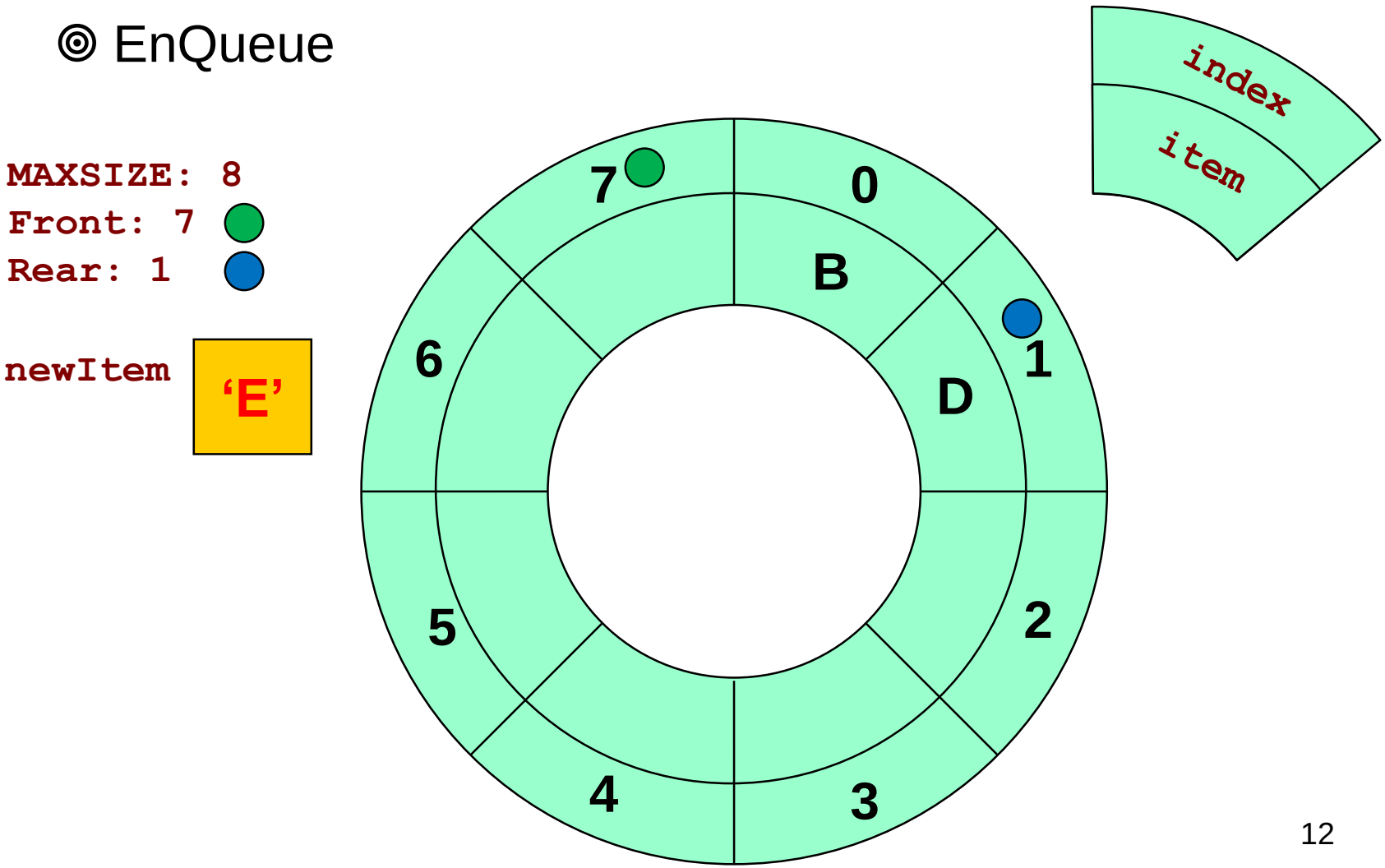
Lab03: Reference (1/3)

◎ EnQueue



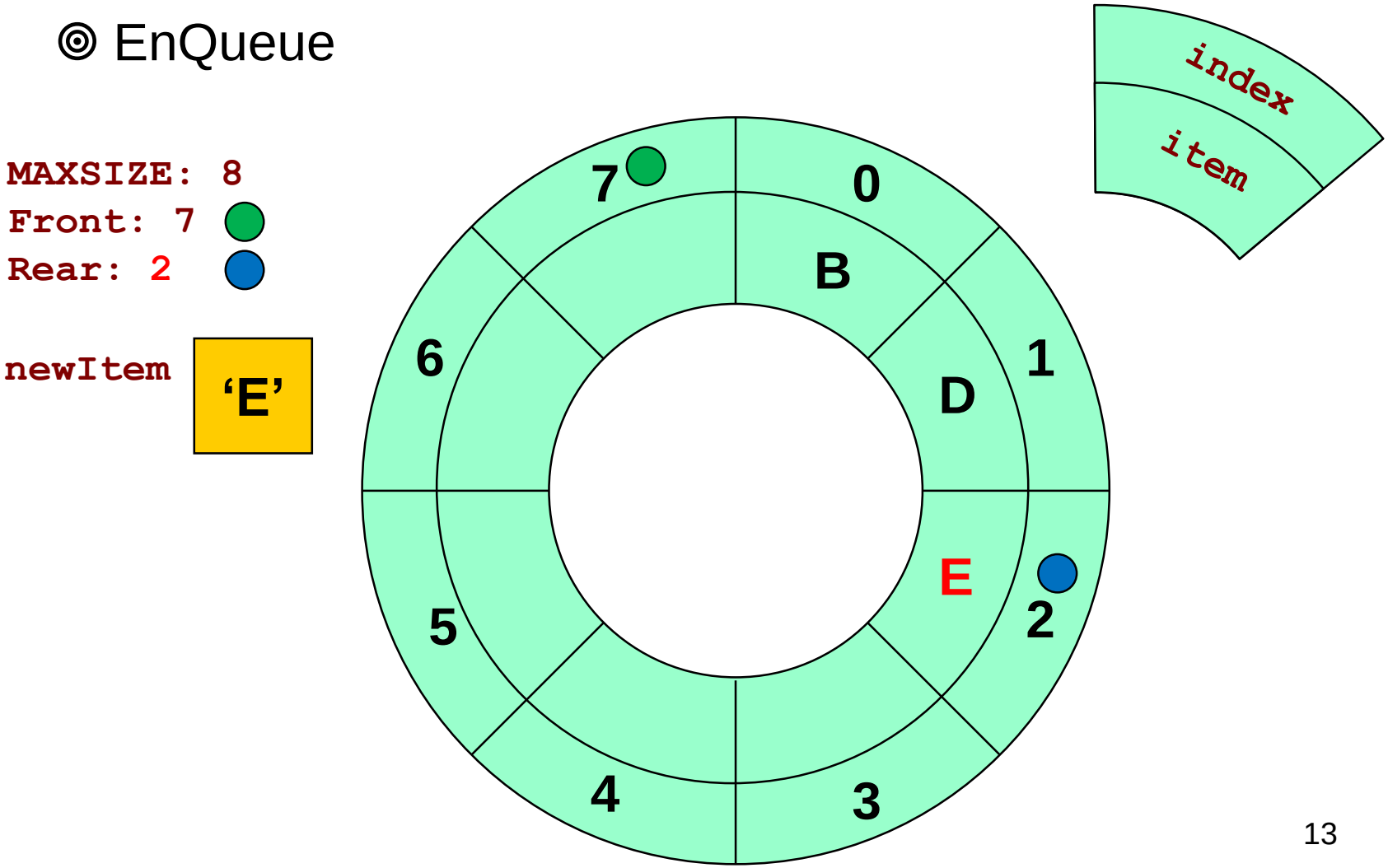
Lab03: Reference (1/3)

◎ EnQueue



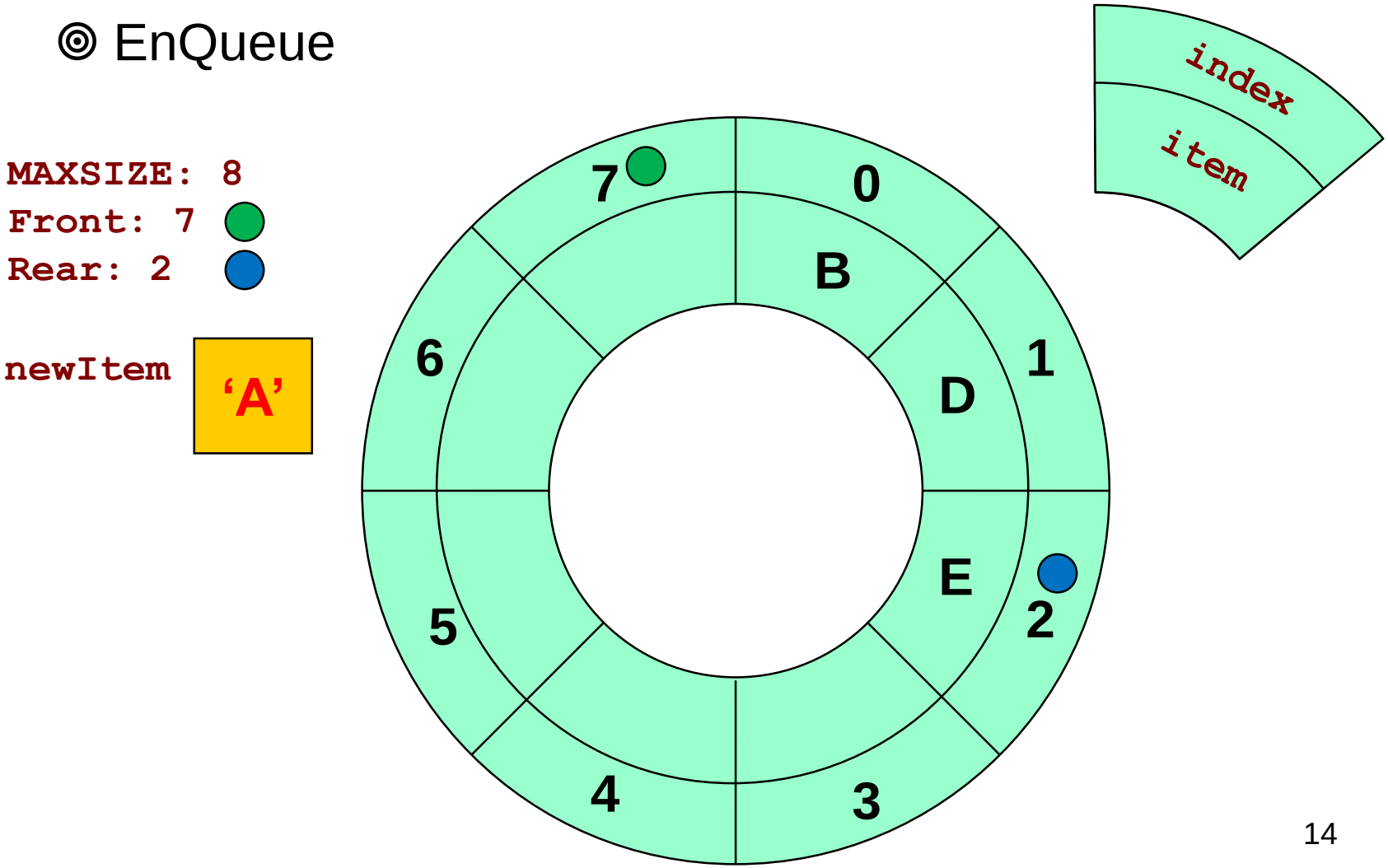
Lab03: Reference (1/3)

◎ EnQueue



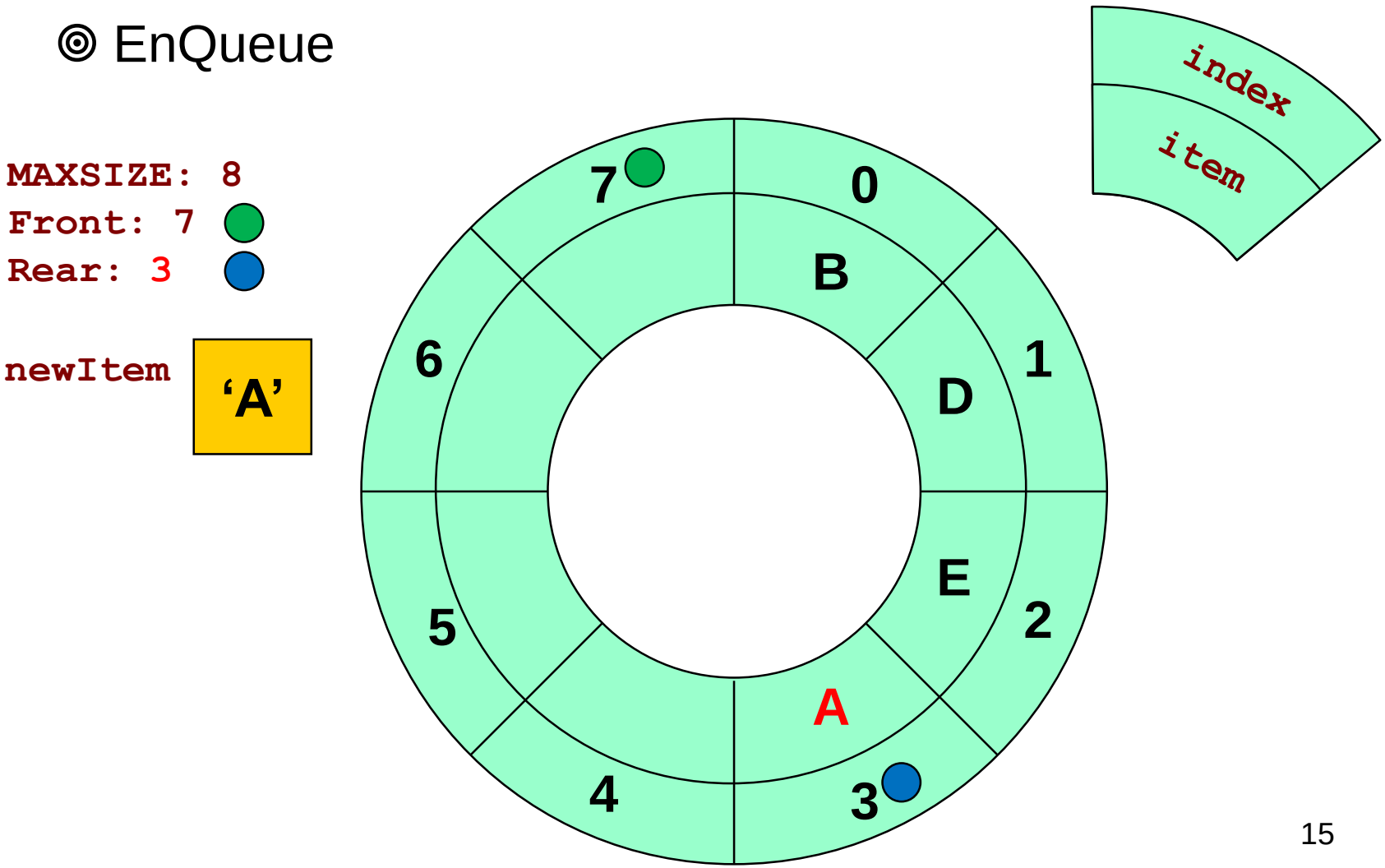
Lab03: Reference (1/3)

◎ EnQueue



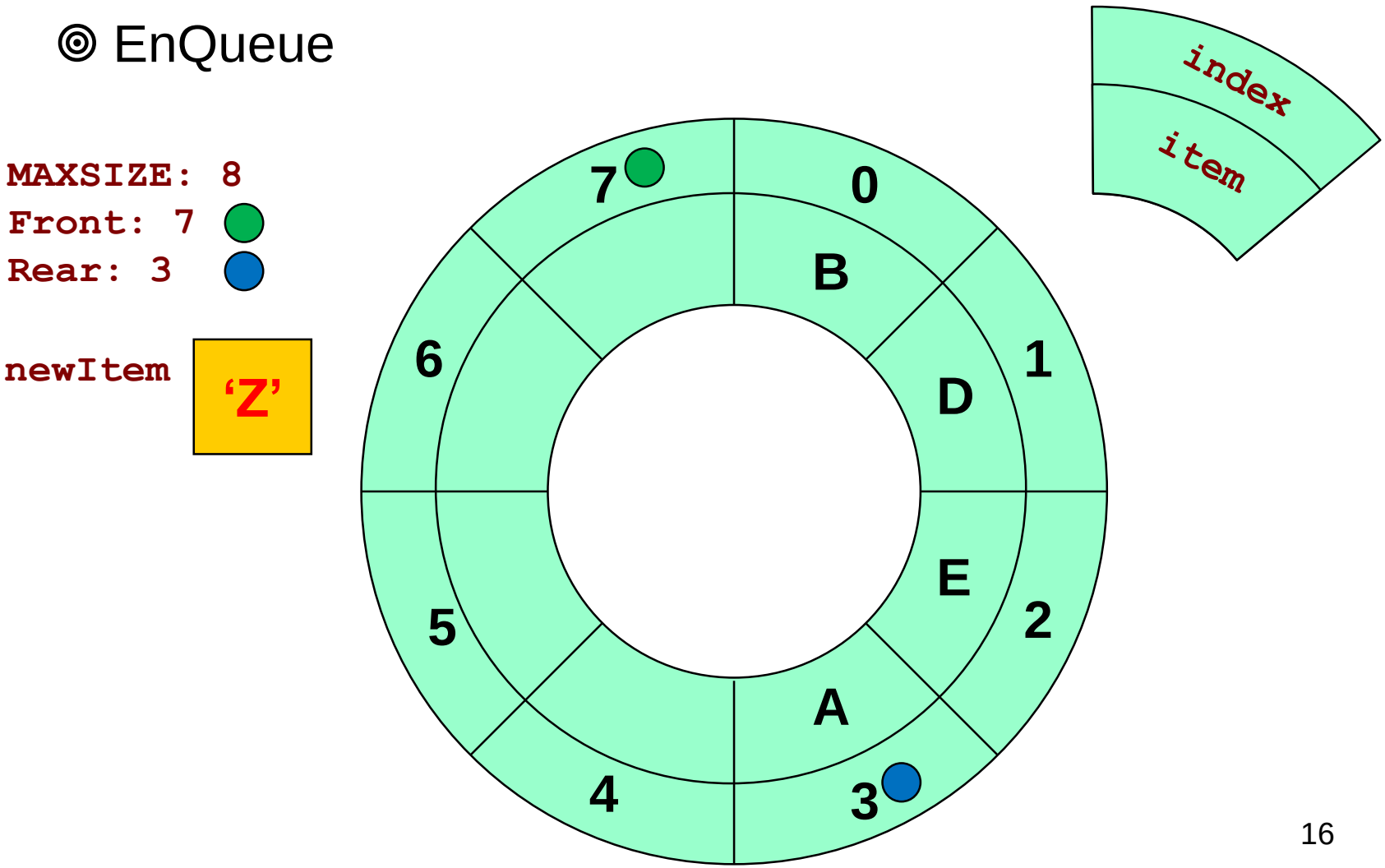
Lab03: Reference (1/3)

◎ EnQueue



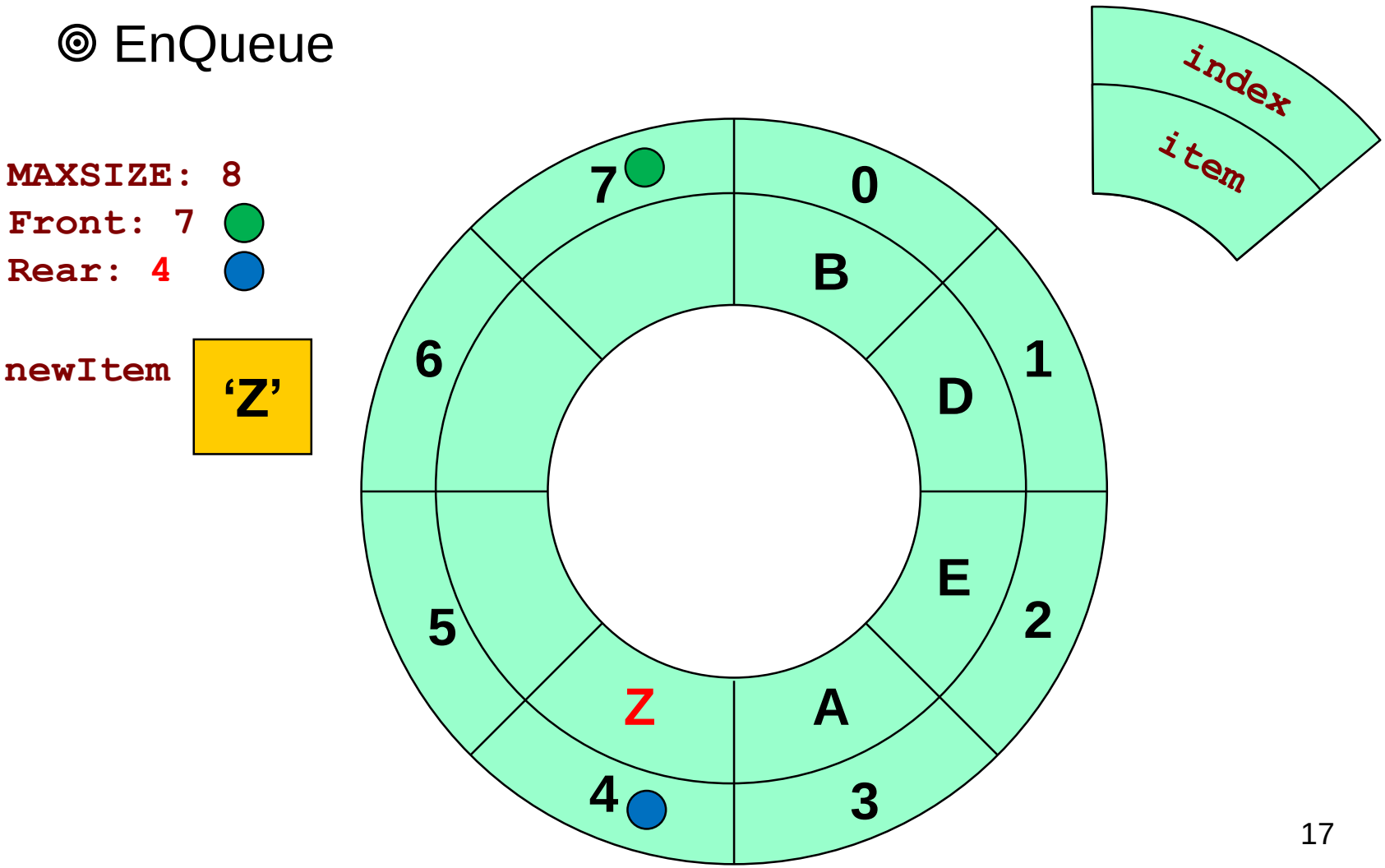
Lab03: Reference (1/3)

◎ EnQueue



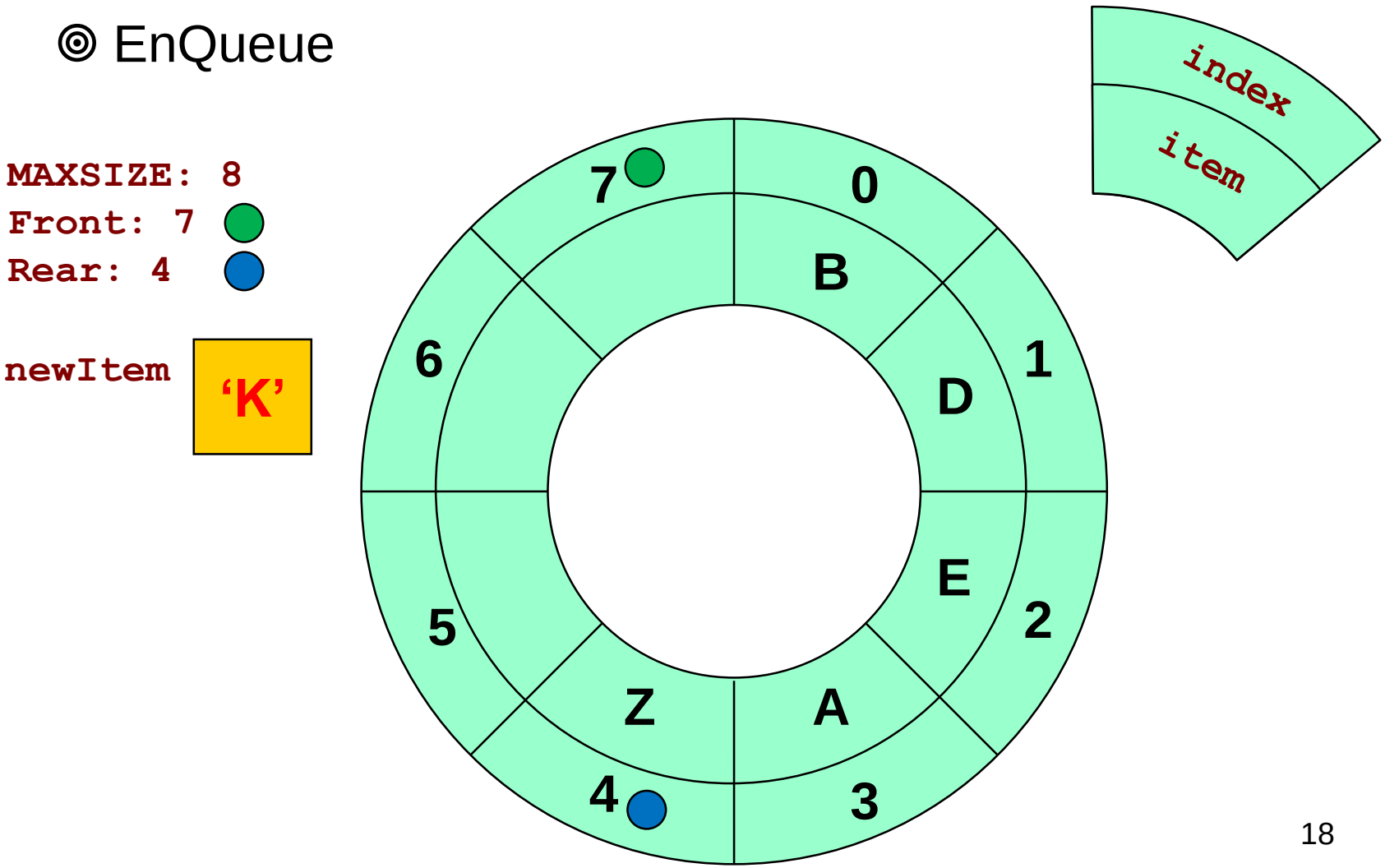
Lab03: Reference (1/3)

◎ EnQueue



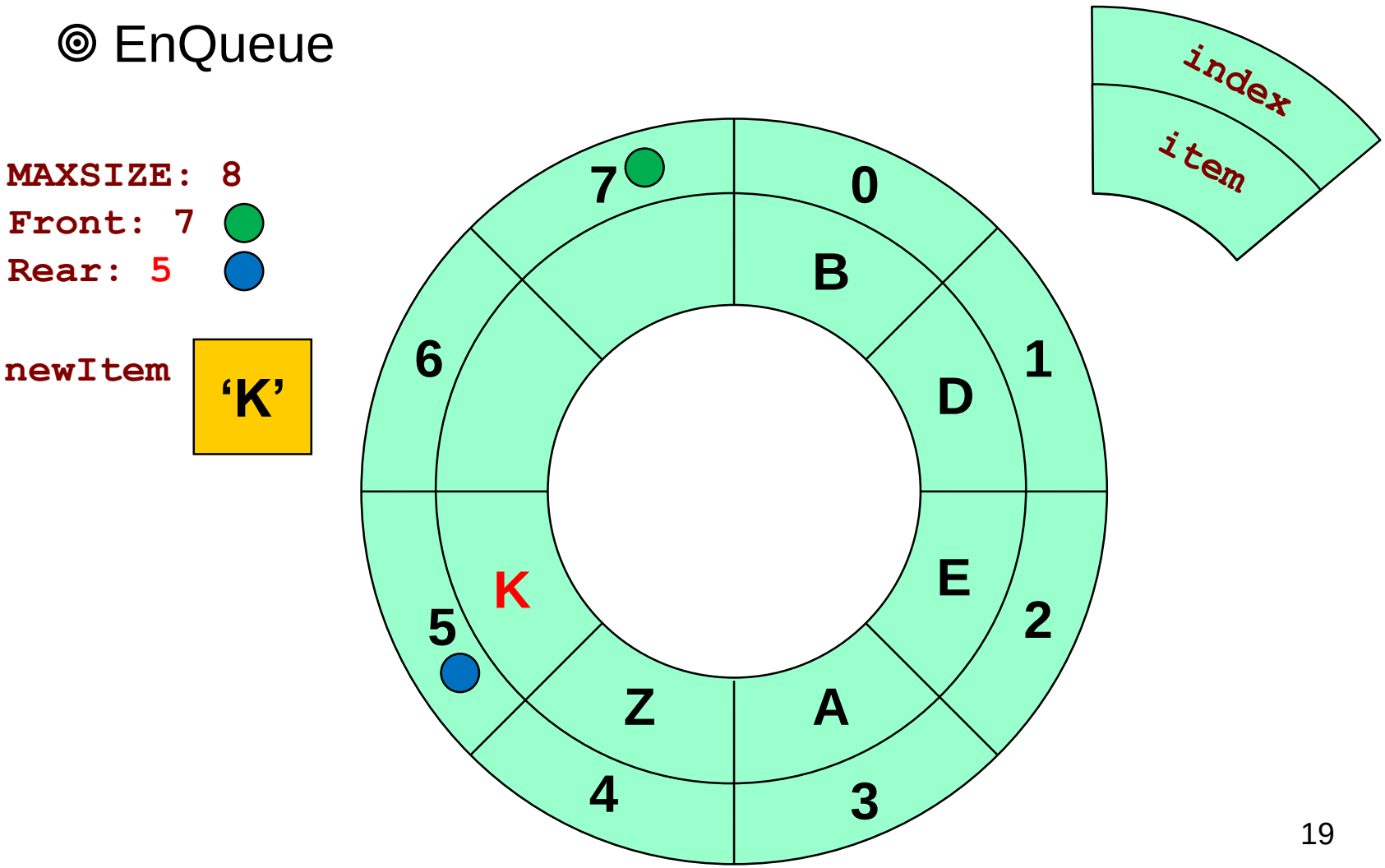
Lab03: Reference (1/3)

◎ EnQueue



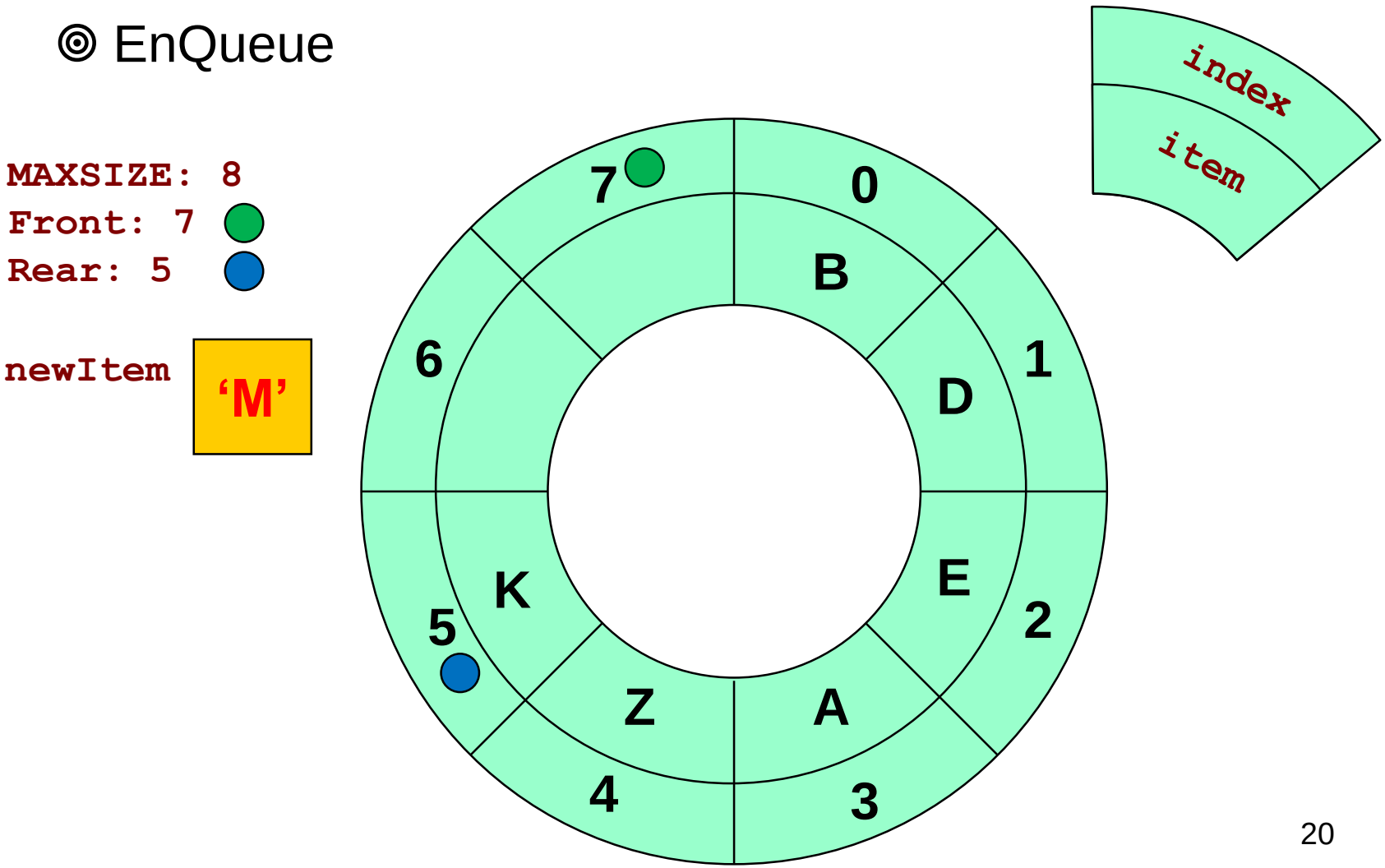
Lab03: Reference (1/3)

◎ EnQueue



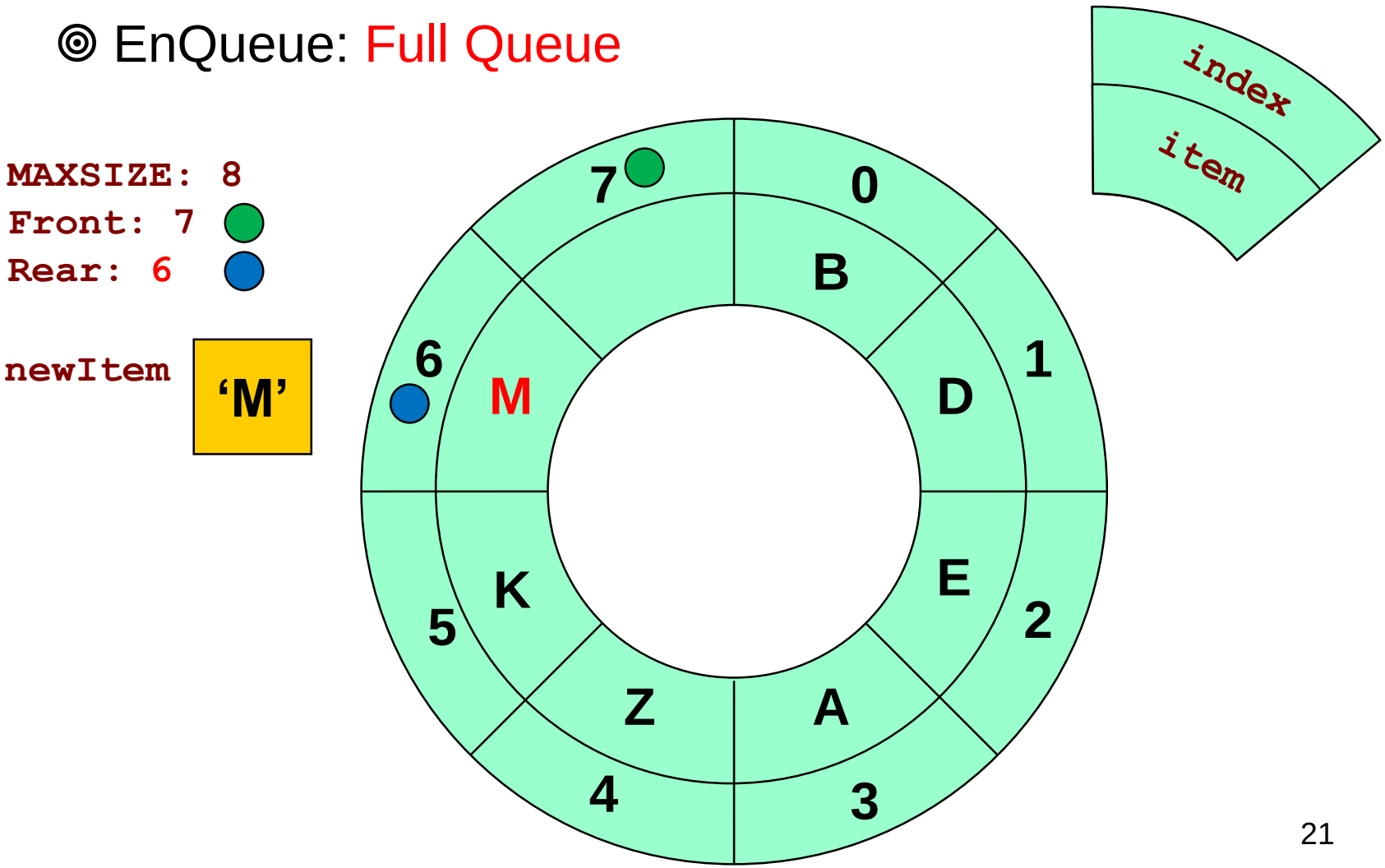
Lab03: Reference (1/3)

◎ EnQueue



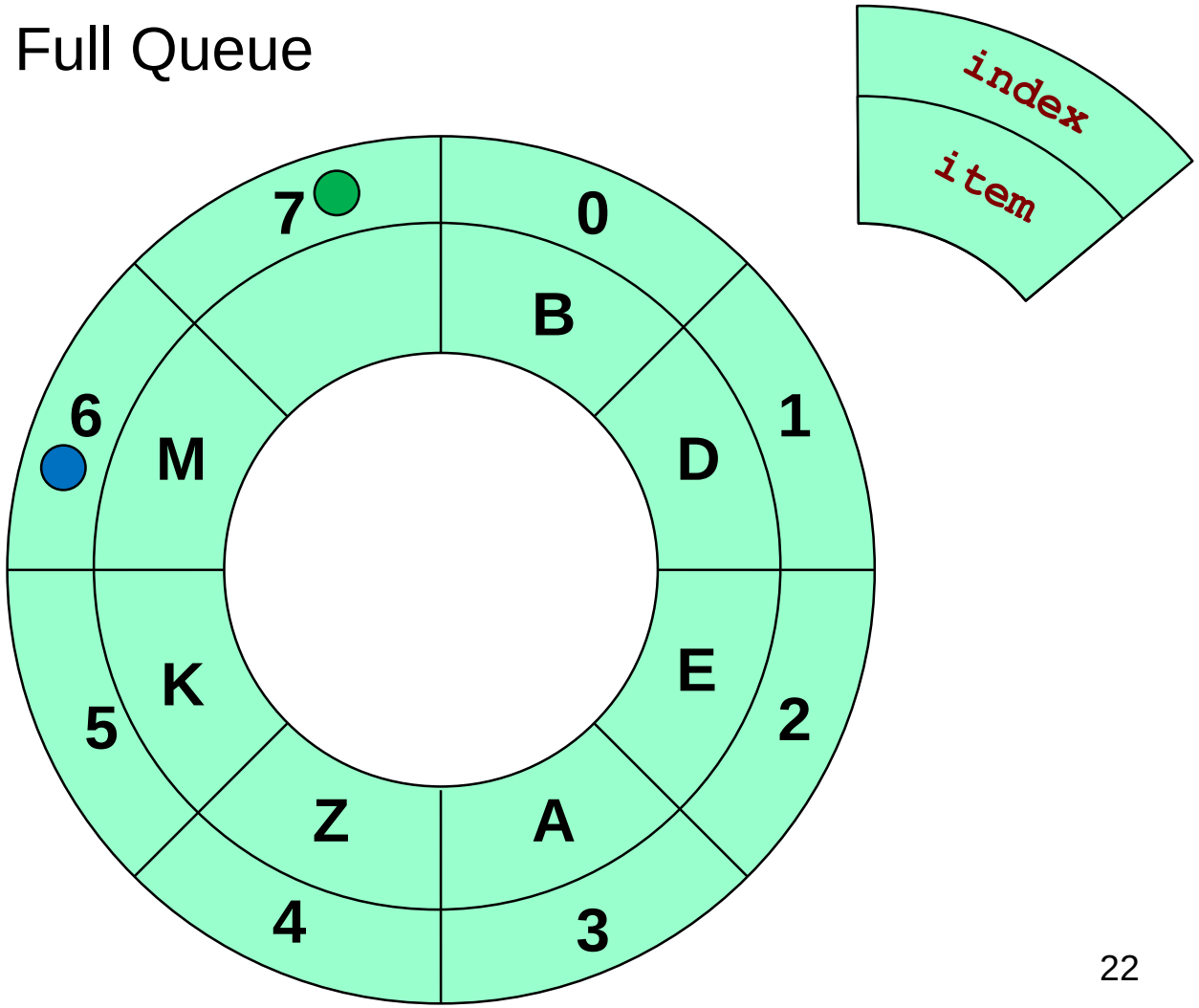
Lab03: Reference (1/3)

◎ EnQueue: Full Queue



Lab03: Reference (2/3)

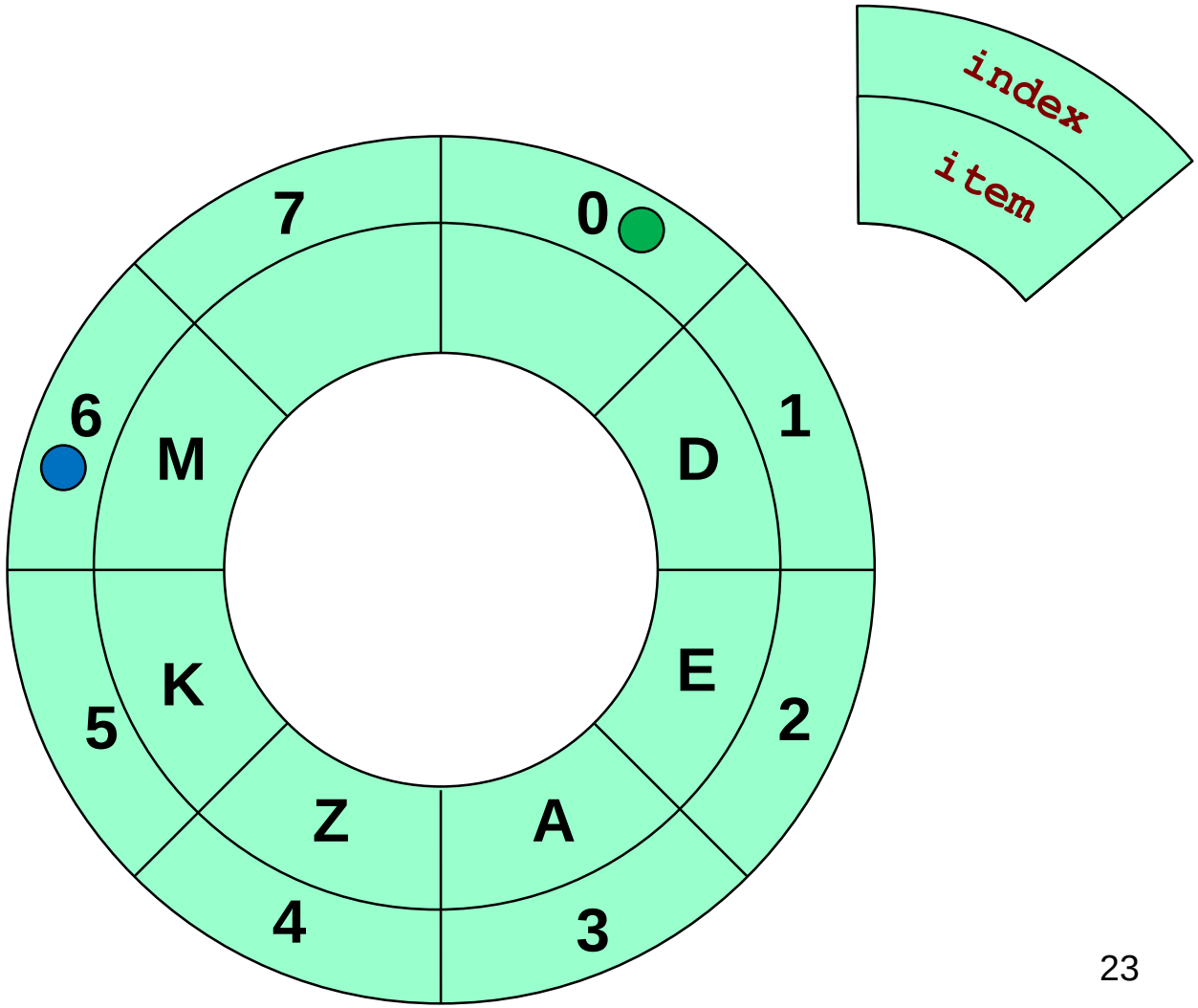
◎ DeQueue: Full Queue



Lab03: Reference (2/3)

◎ DeQueue

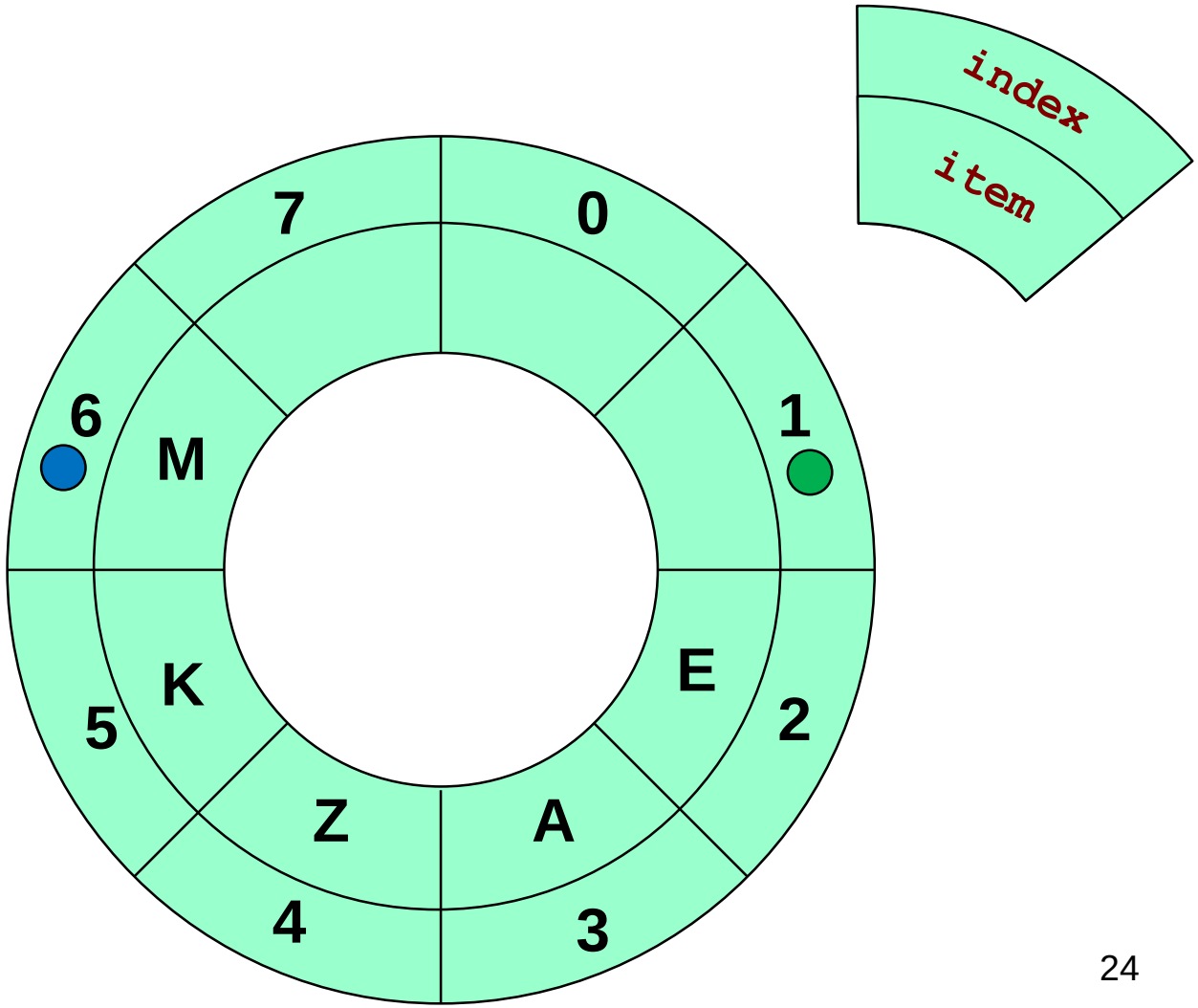
MAXSIZE: 8
Front: 0
Rear: 6



Lab03: Reference (2/3)

◎ DeQueue

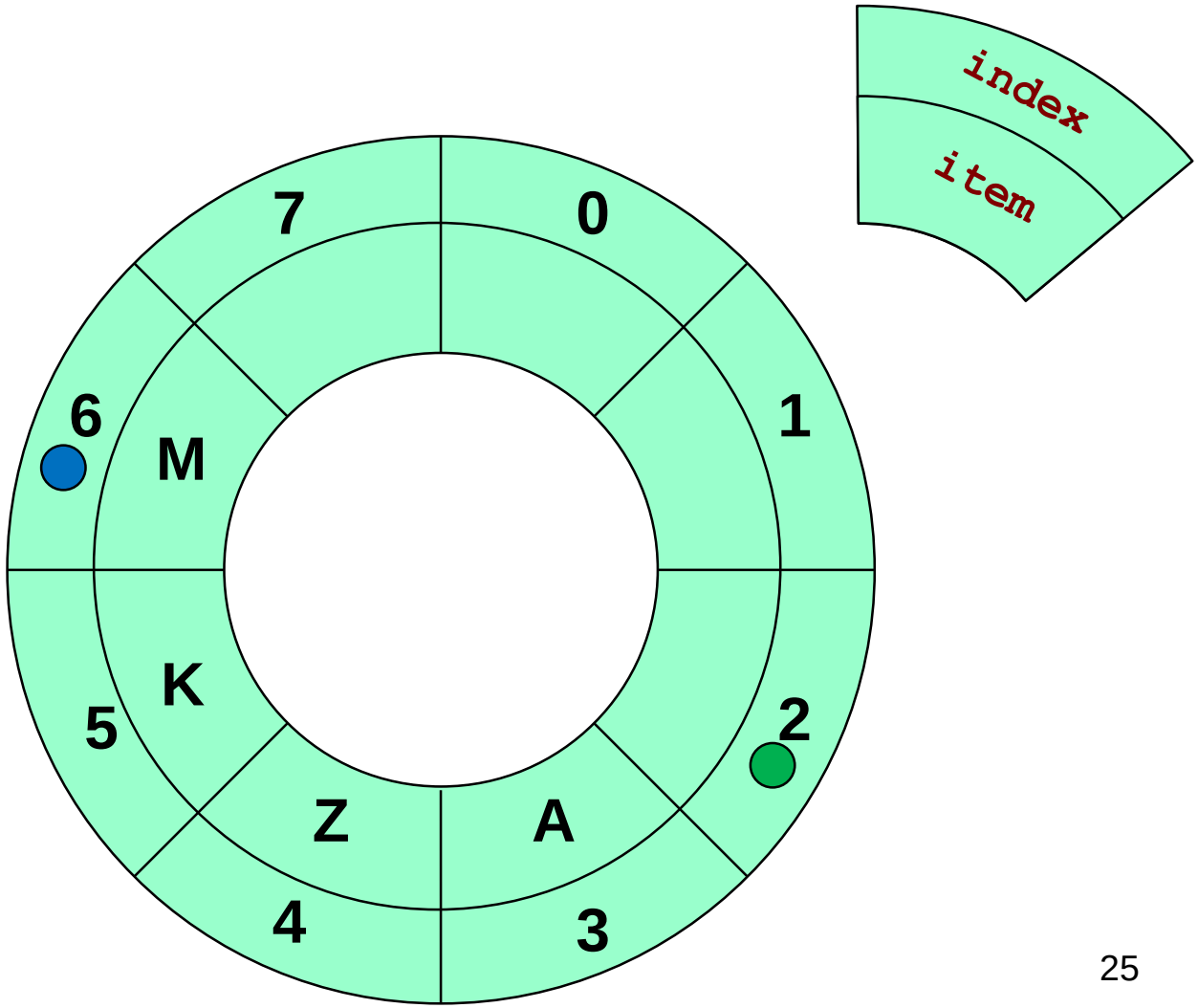
MAXSIZE: 8
Front: 1
Rear: 6



Lab03: Reference (2/3)

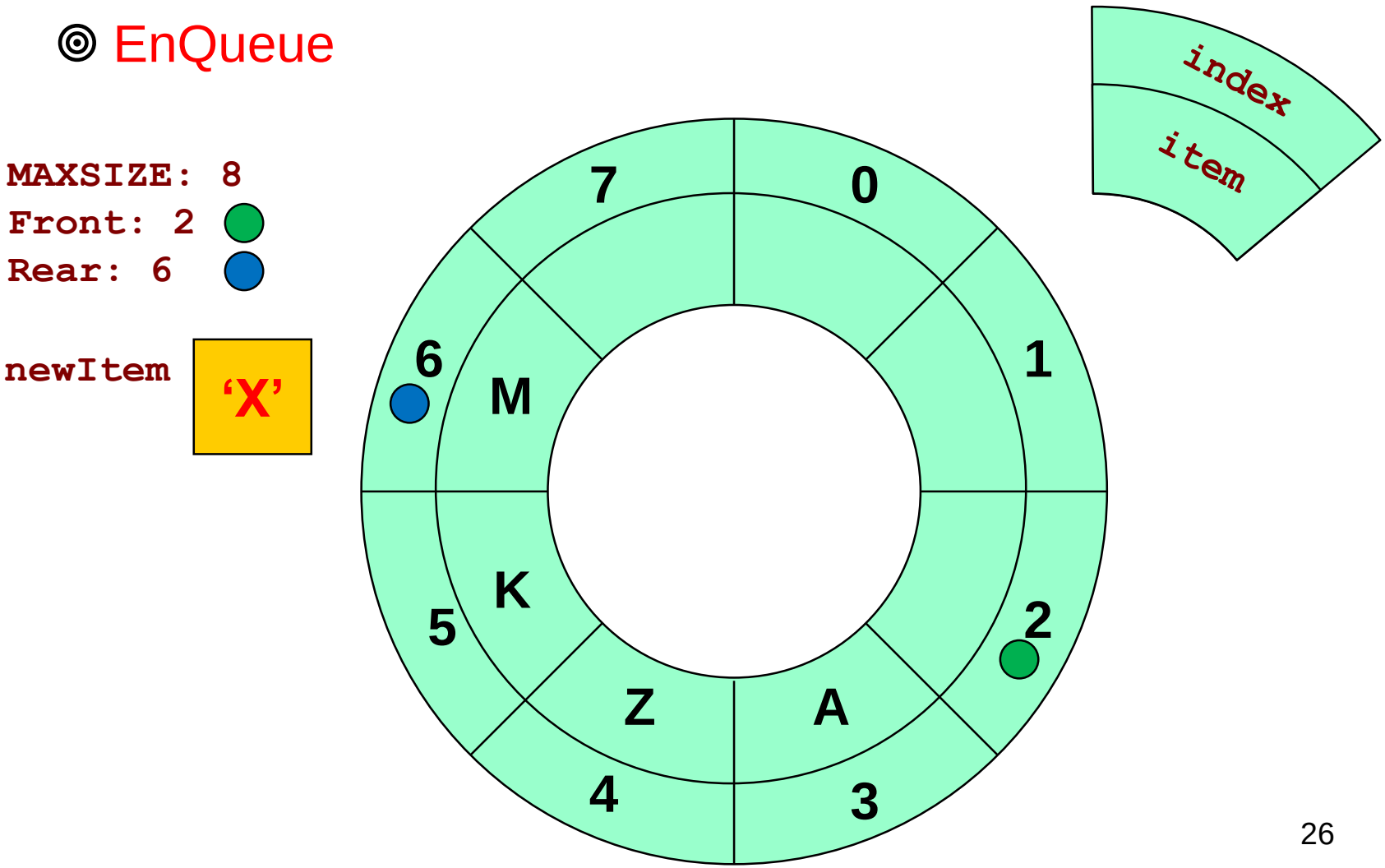
◎ DeQueue

MAXSIZE: 8
Front: 2
Rear: 6



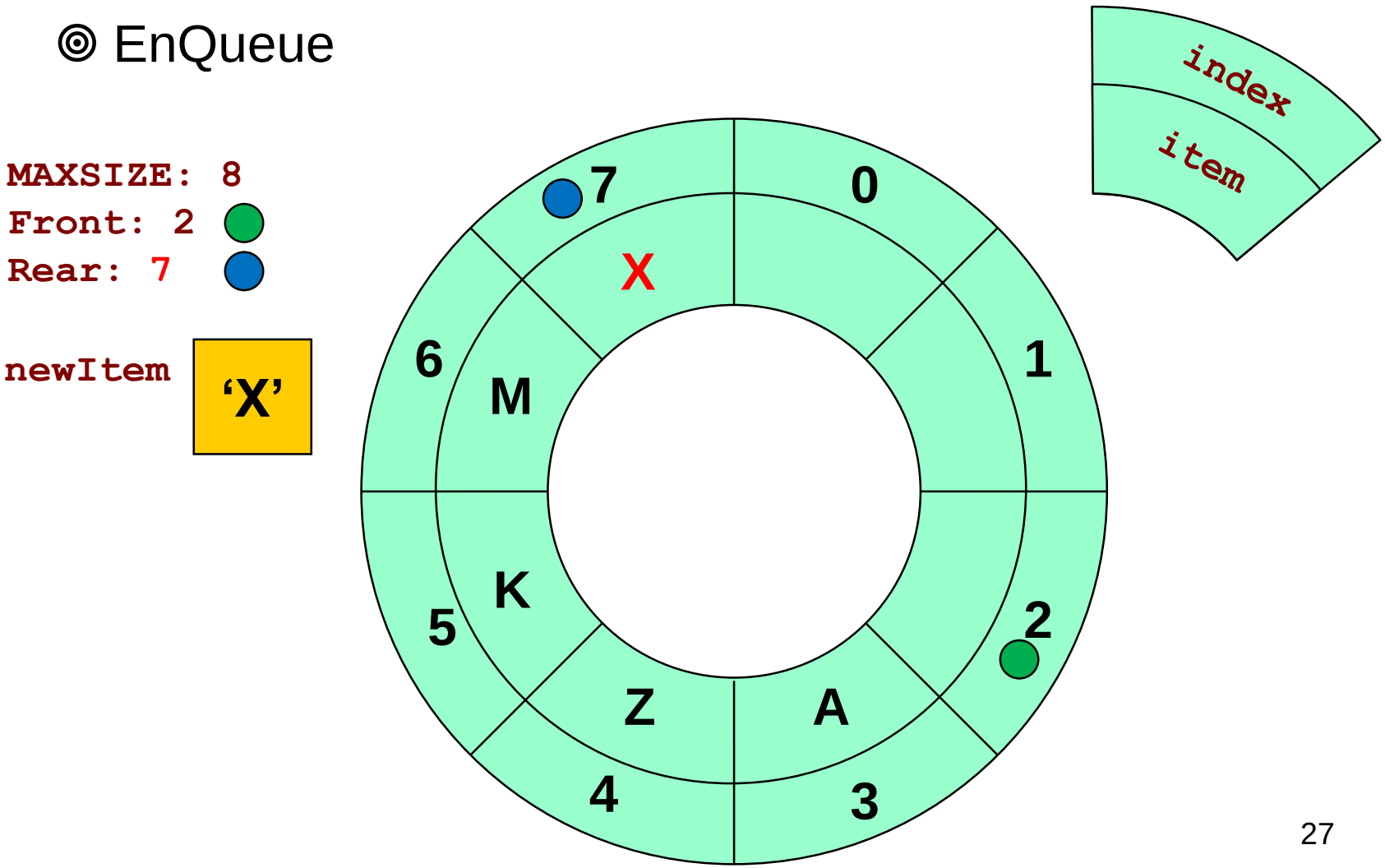
Lab03: Reference (3/3)

- EnQueue



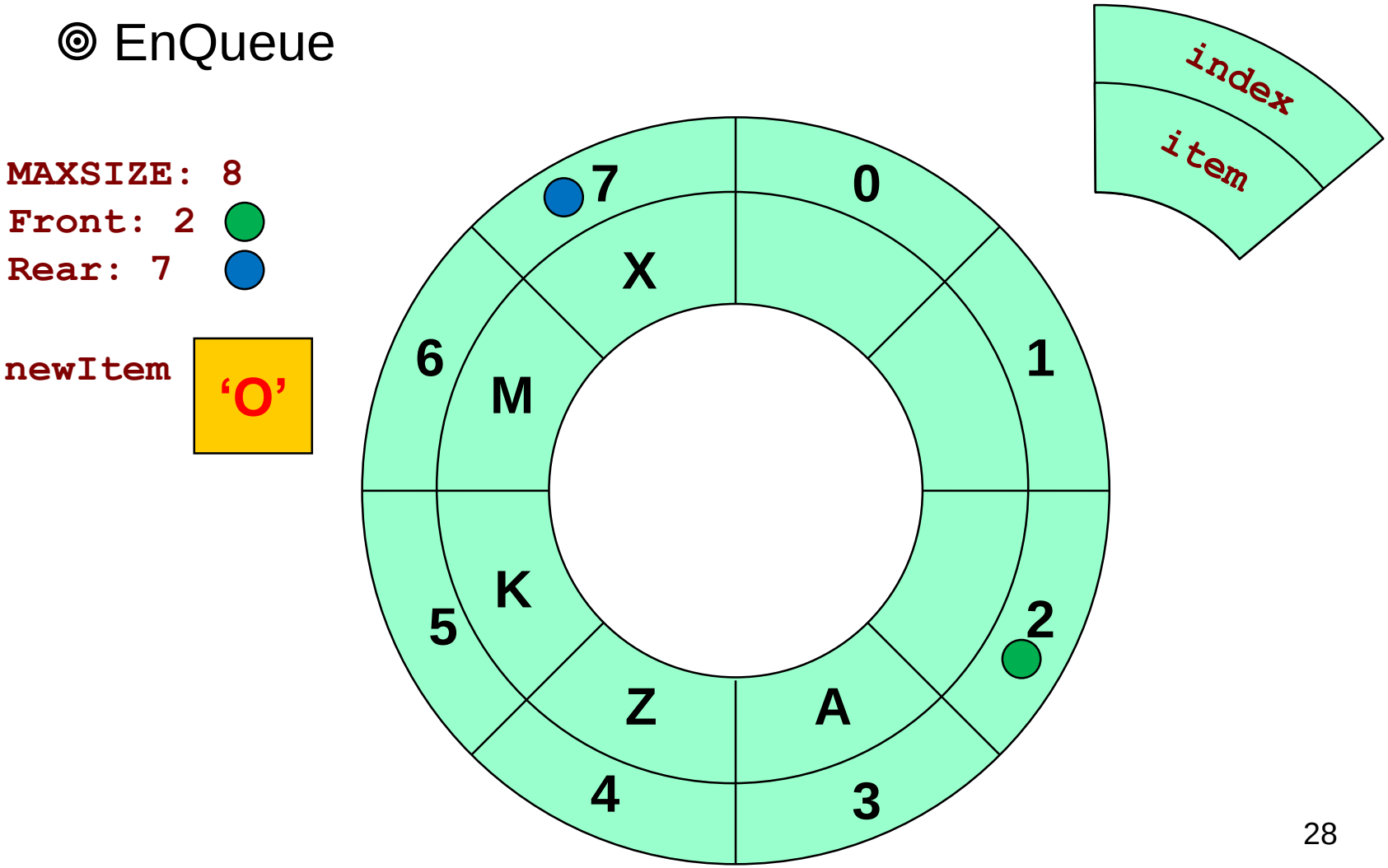
Lab03: Reference (3/3)

◎ EnQueue



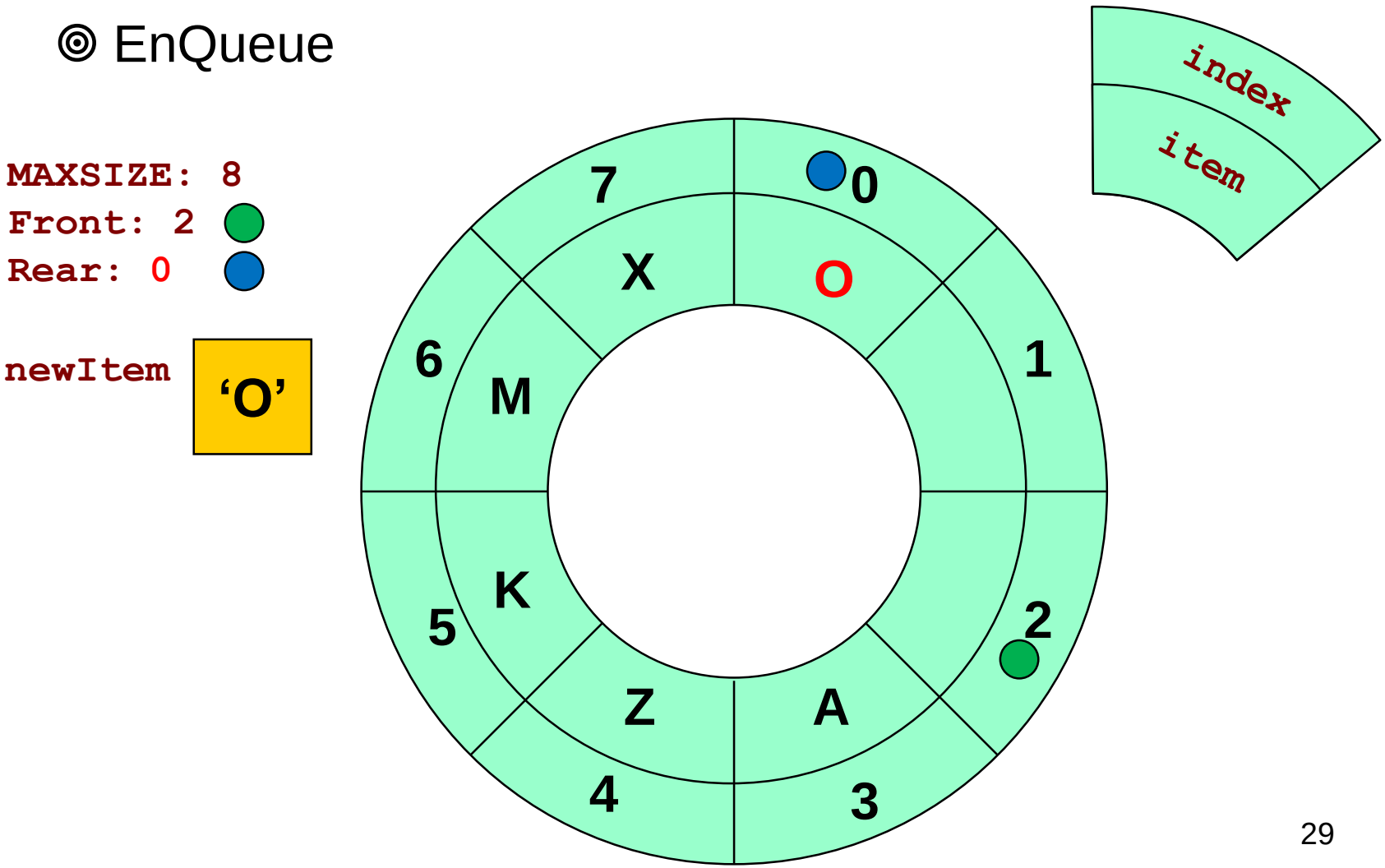
Lab03: Reference (3/3)

◎ EnQueue



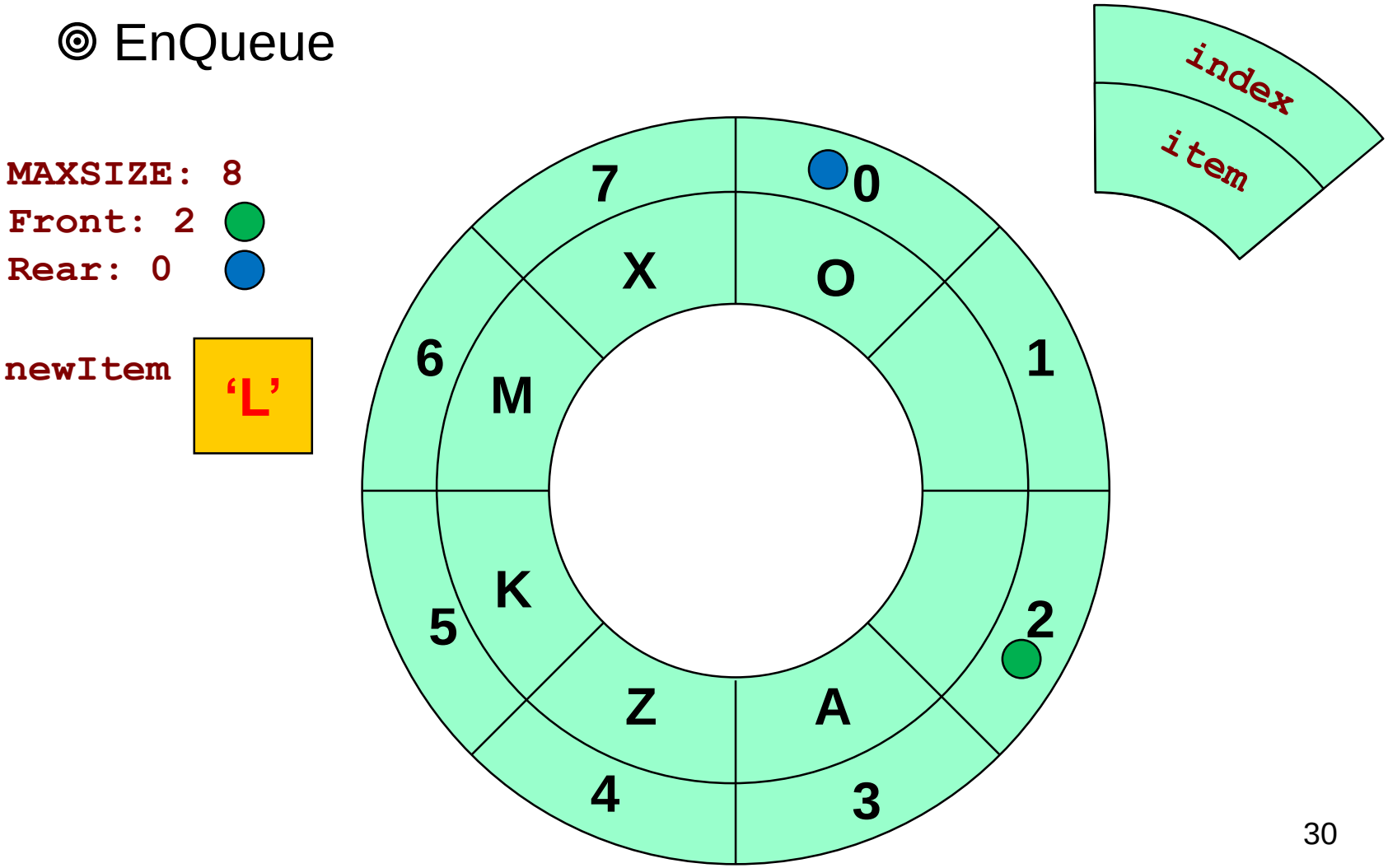
Lab03: Reference (3/3)

◎ EnQueue



Lab03: Reference (3/3)

◎ EnQueue



Lab03: Reference (3/3)

◎ EnQueue: Full Queue