자료구조 실습03

Data Structures Lab03

Lab03 예제(1/2)

◎ 목표: Template을 이용한 Queue class 설계 및 구현

◎ 내용:

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

◎ 방법

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

Lab03 예제(2/2)

◎ 내용:

- ☞ Template의 사용 방법과 Queue의 동작 원리를 파악하여 작성
- ☞ 주어진 코드를 분석 후 Template 을 이용한 Generic Queue를 작성

◎ 방법

- ☞ 주어진 Stack 소스코드를 분석한 후 char, int, string 등 다양한 자료형을 이용할 수 있도록 Template 기반의 Circular Queue Application 작성

예제: 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
};
```

예제: console

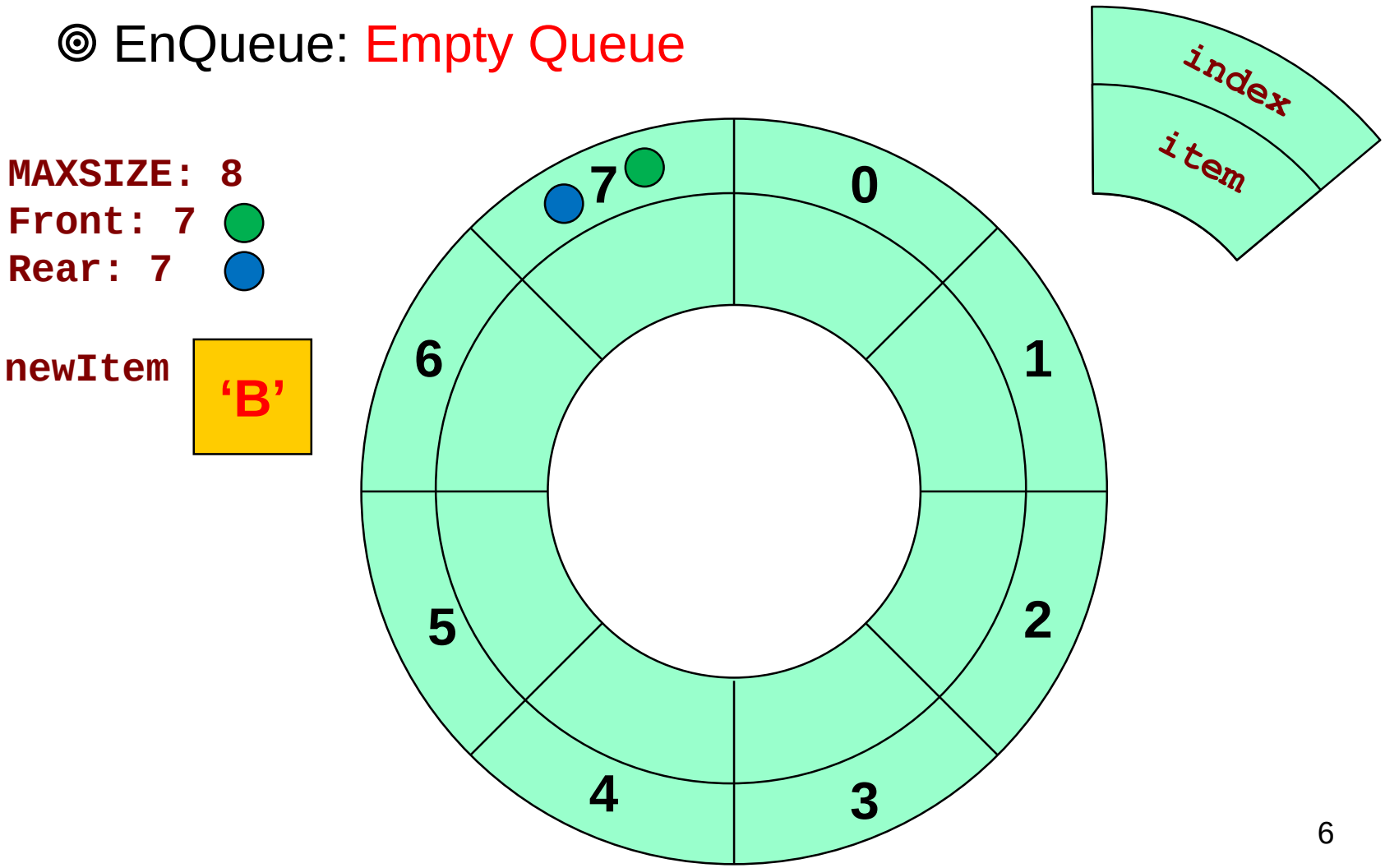
◎ Queue를 테스트할 driver는 다음과 같이 작성함

```
--- 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 -->
```

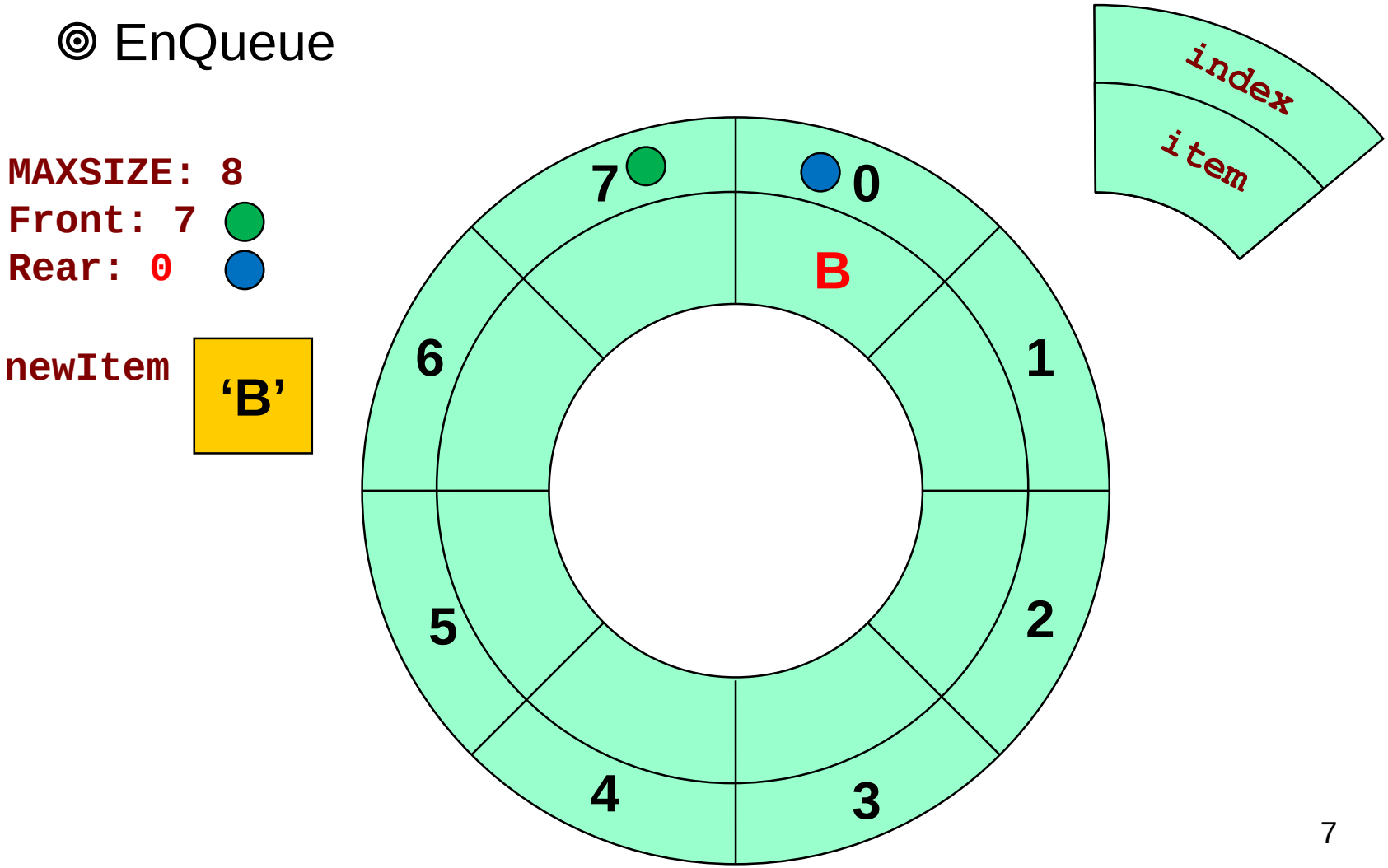
Lab03: Reference (1/3)

◎ EnQueue: Empty Queue



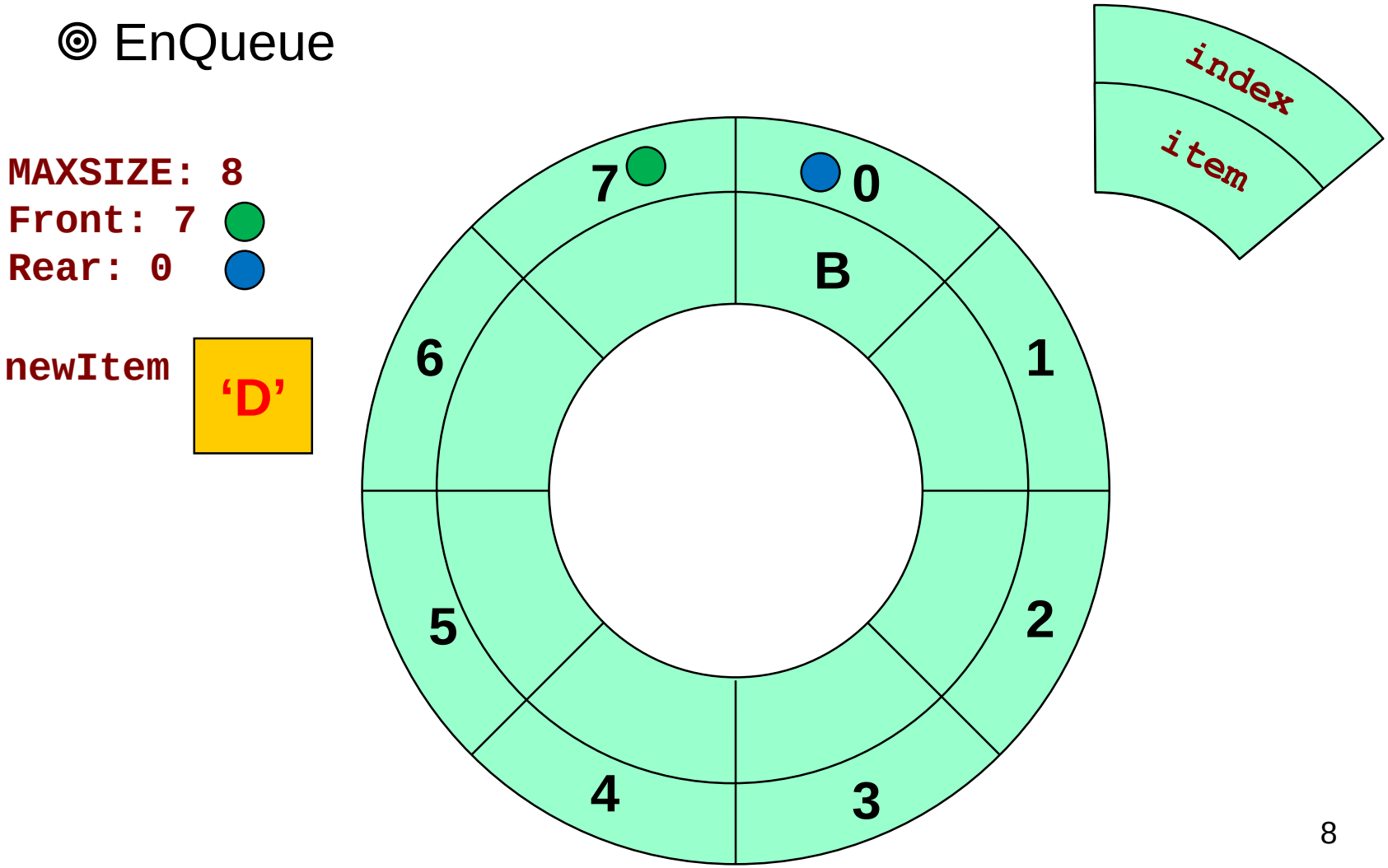
Lab03: Reference (1/3)

◎ EnQueue



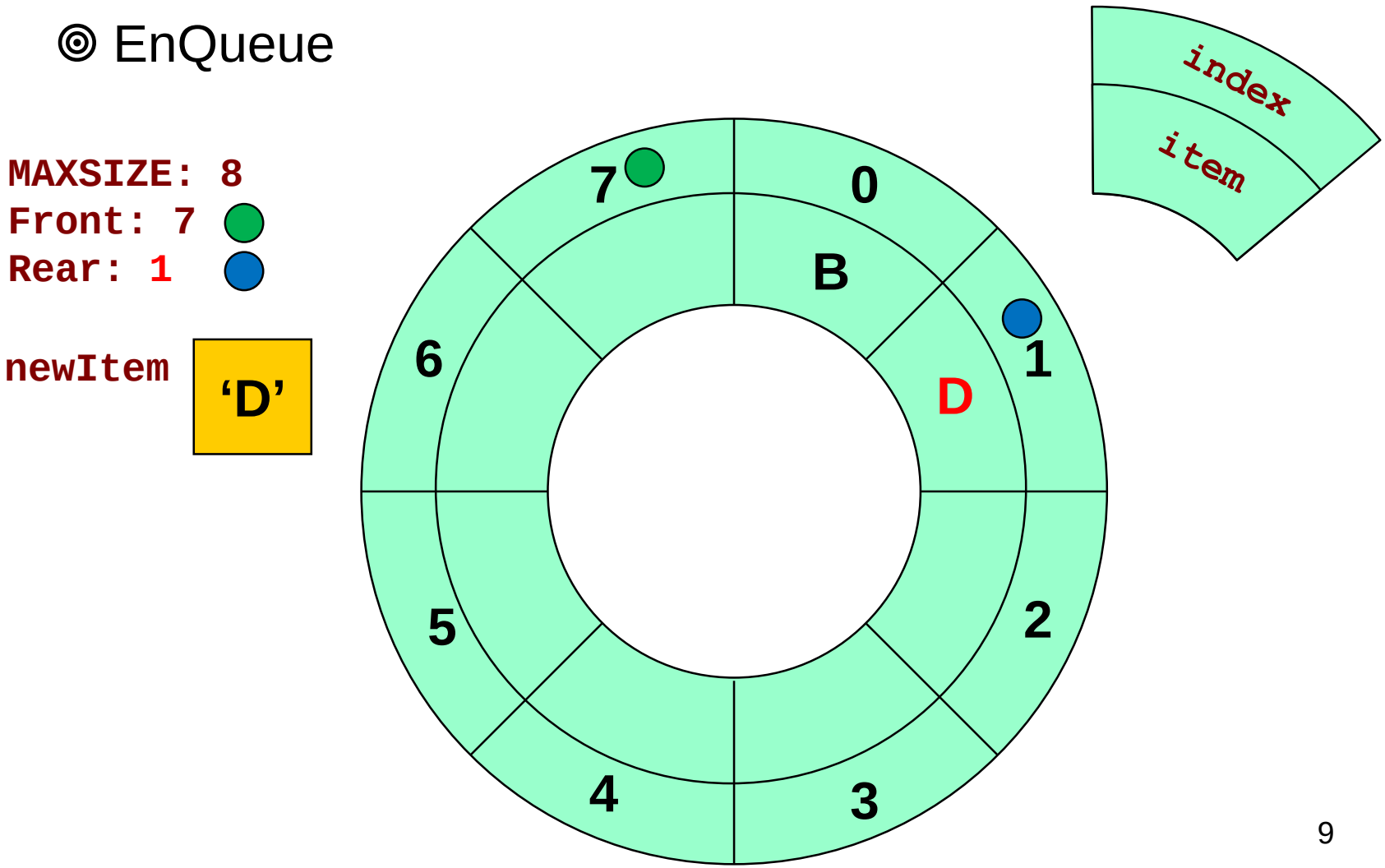
Lab03: Reference (1/3)

◎ EnQueue



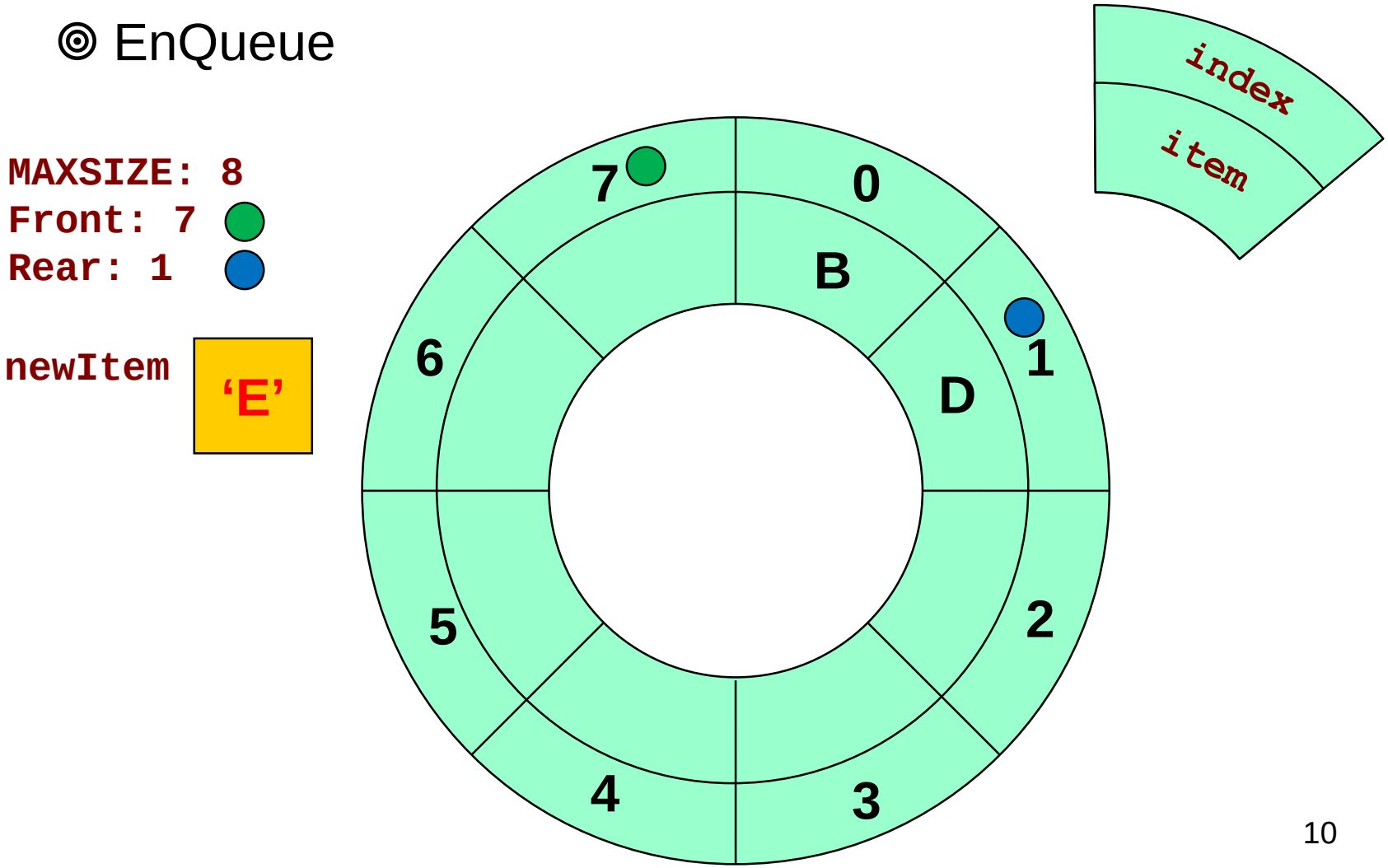
Lab03: Reference (1/3)

◎ EnQueue



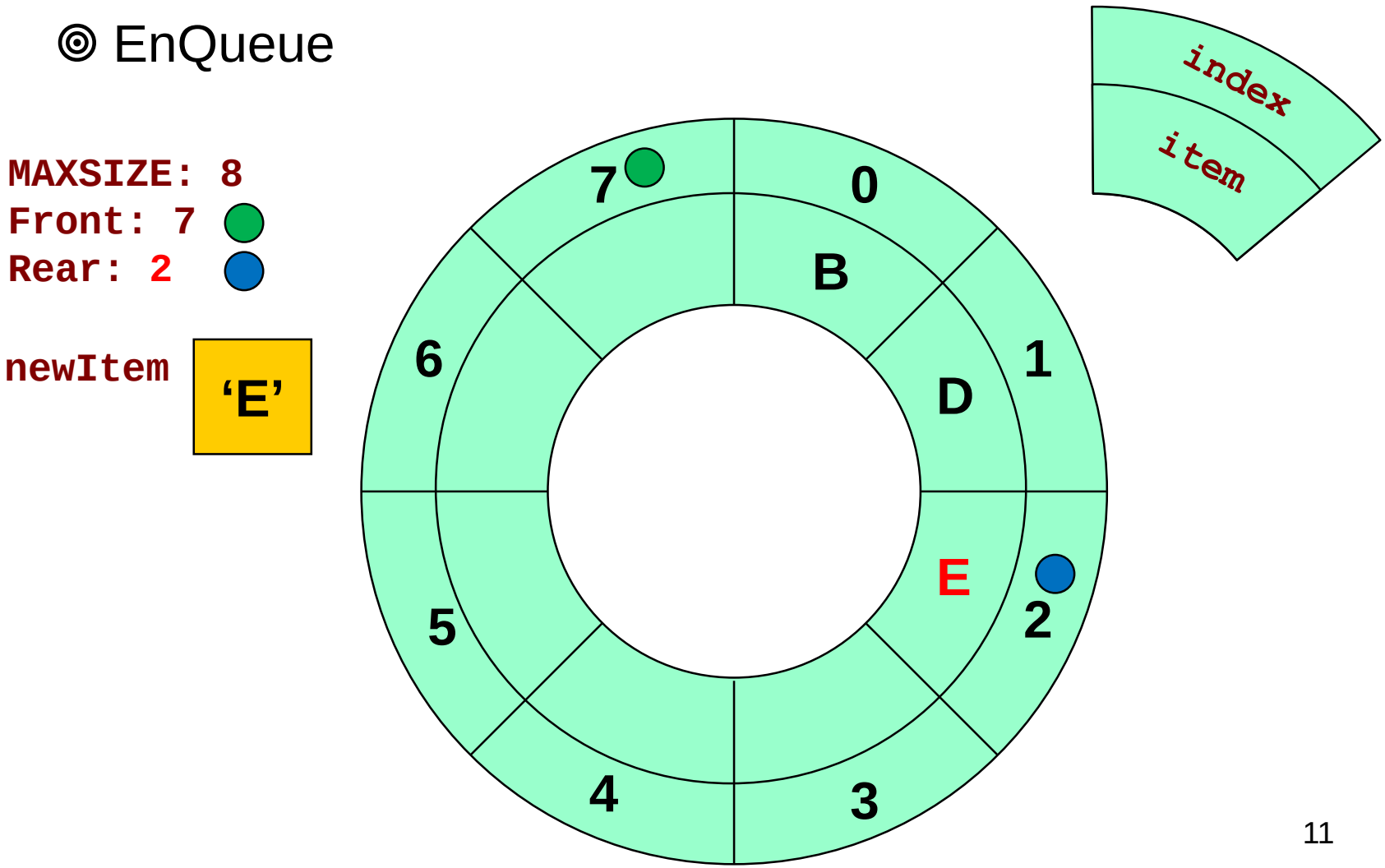
Lab03: Reference (1/3)

◎ EnQueue



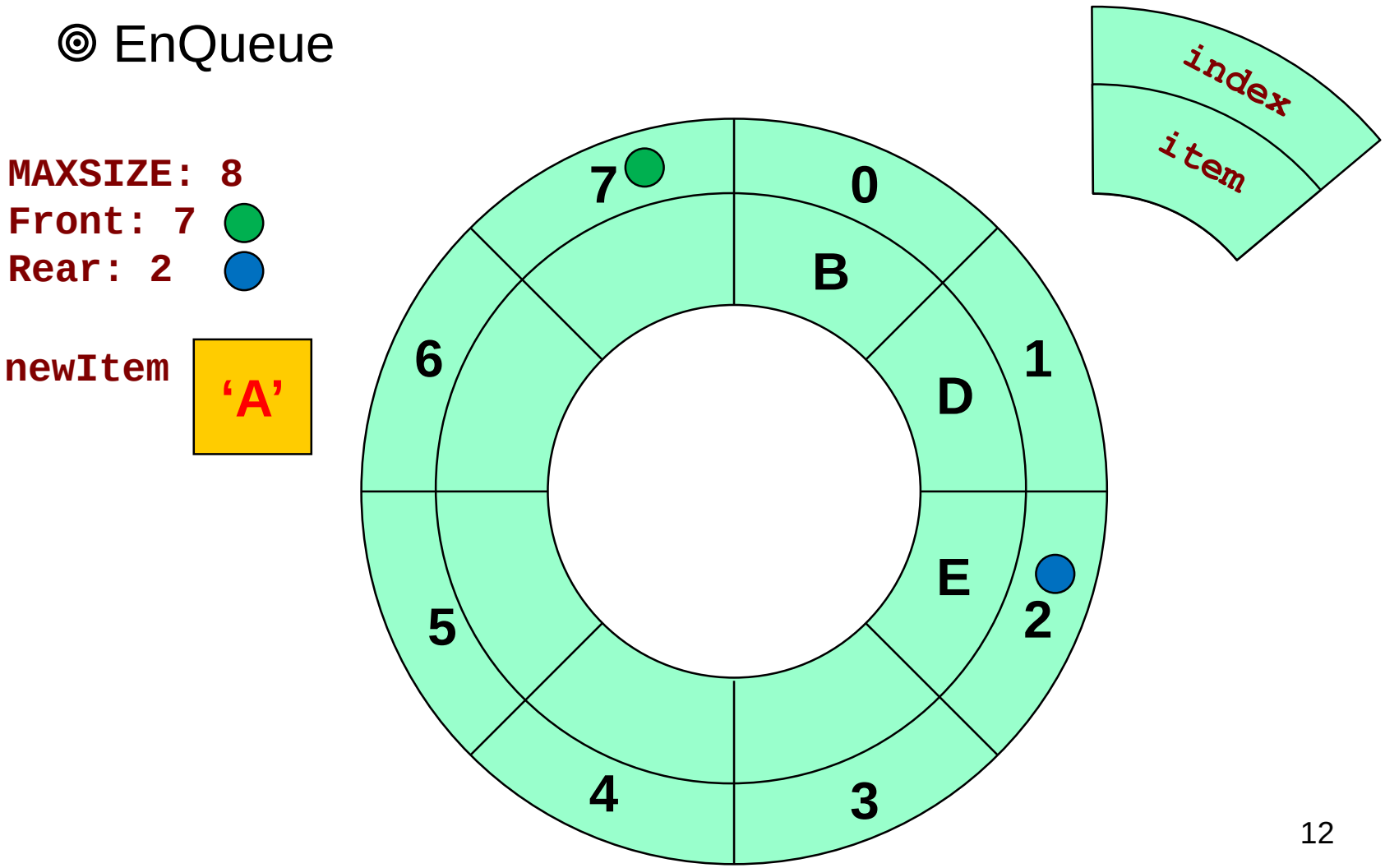
Lab03: Reference (1/3)

◎ EnQueue



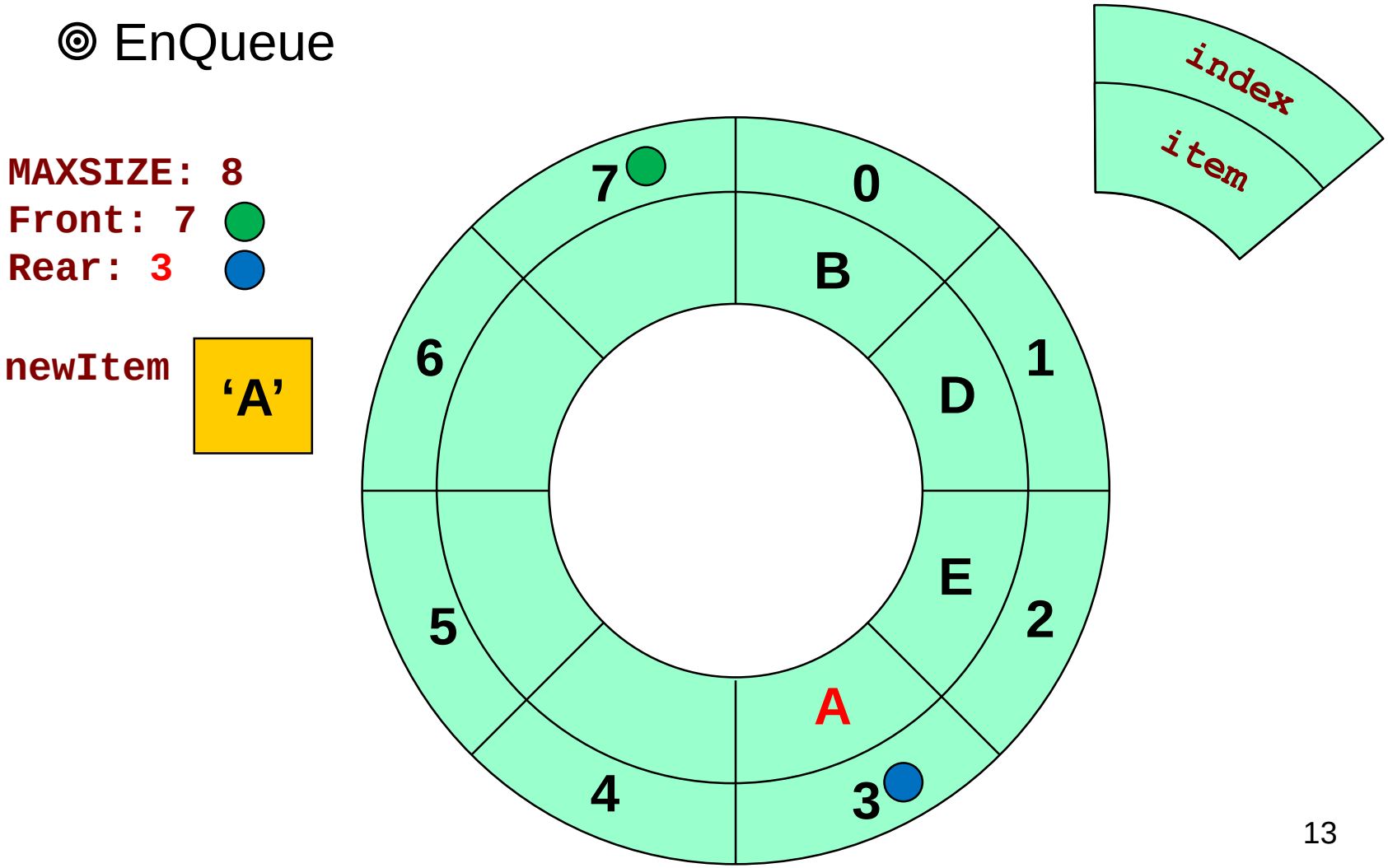
Lab03: Reference (1/3)

◎ EnQueue



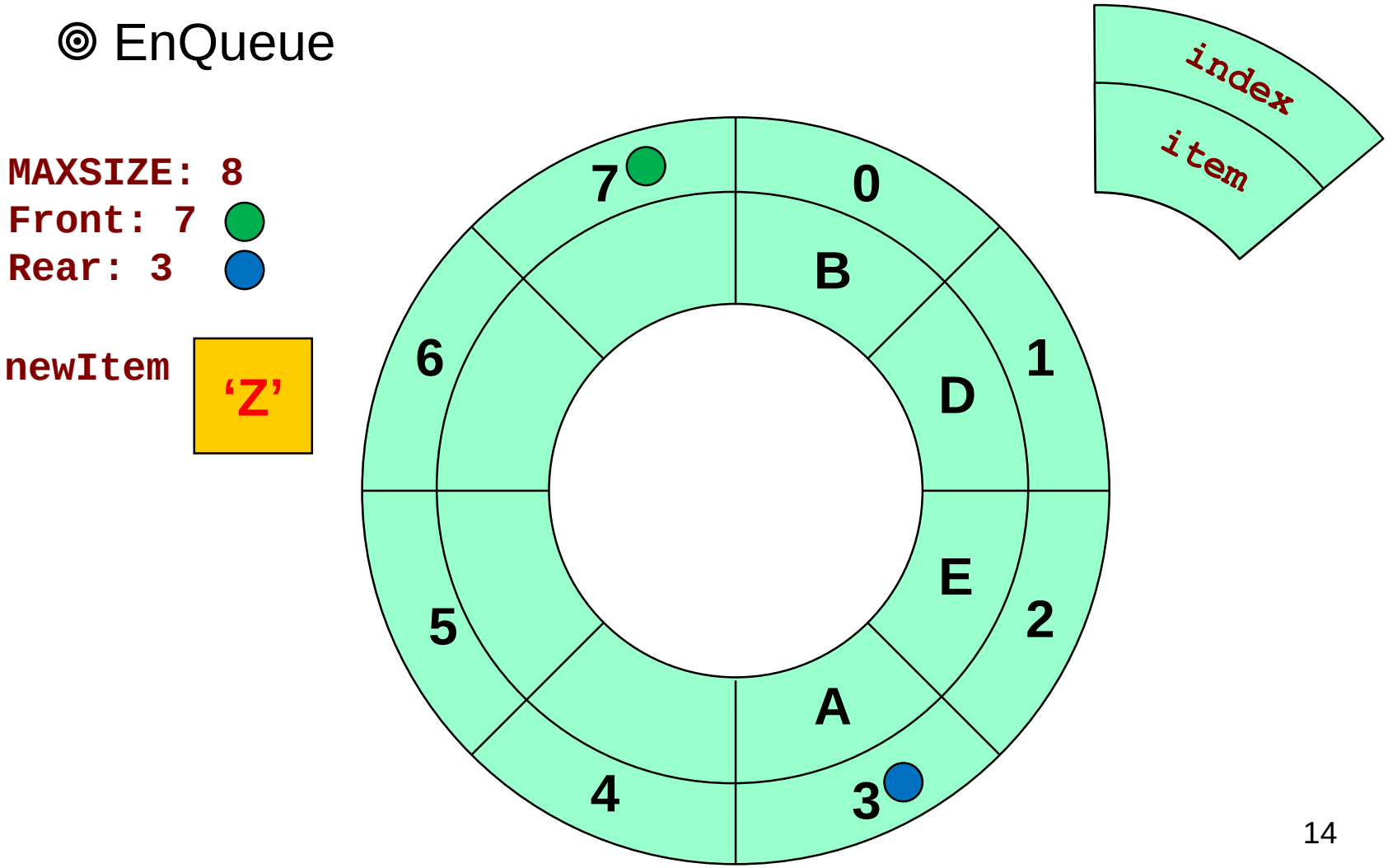
Lab03: Reference (1/3)

◎ EnQueue



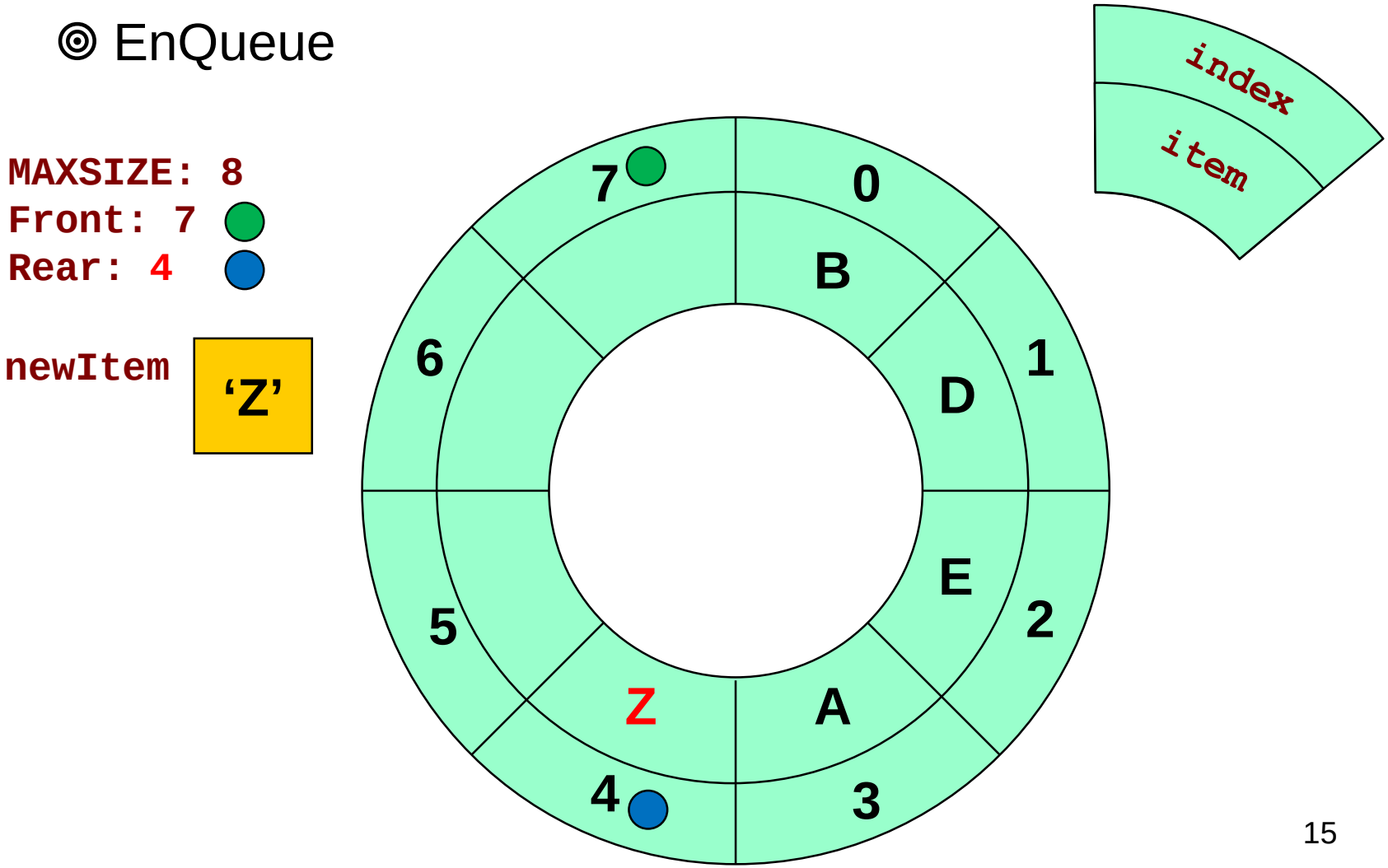
Lab03: Reference (1/3)

◎ EnQueue



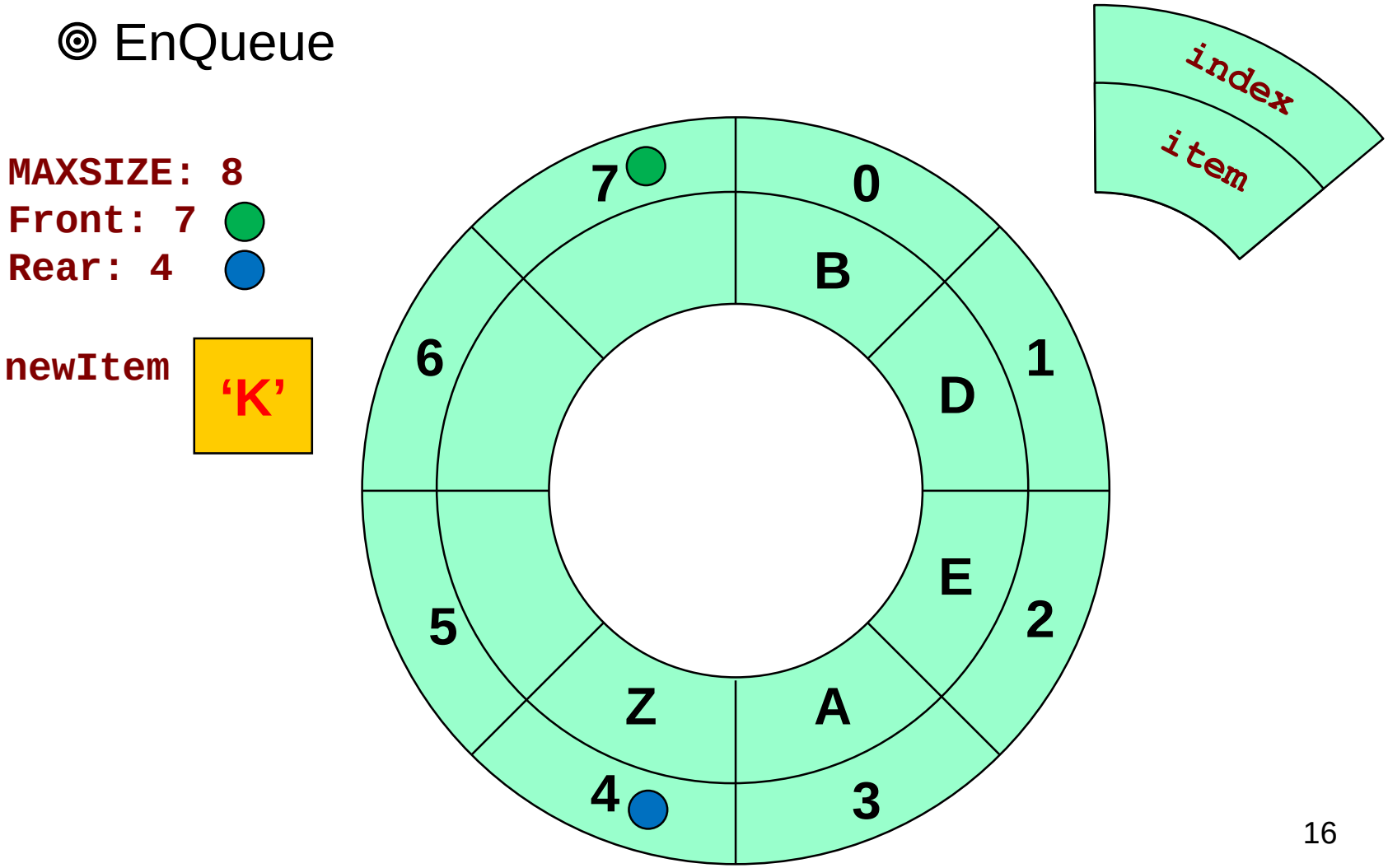
Lab03: Reference (1/3)

◎ EnQueue



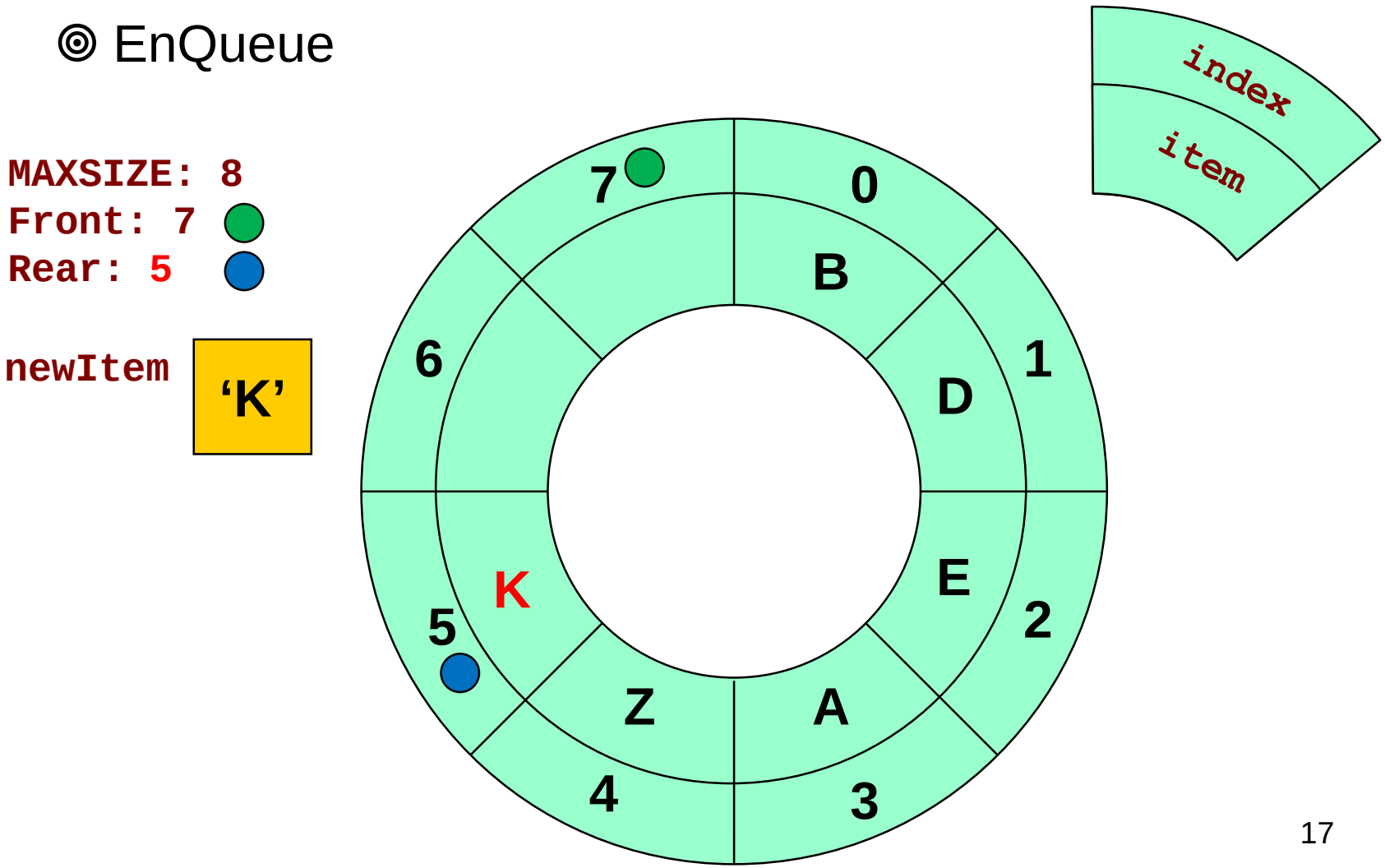
Lab03: Reference (1/3)

◎ EnQueue



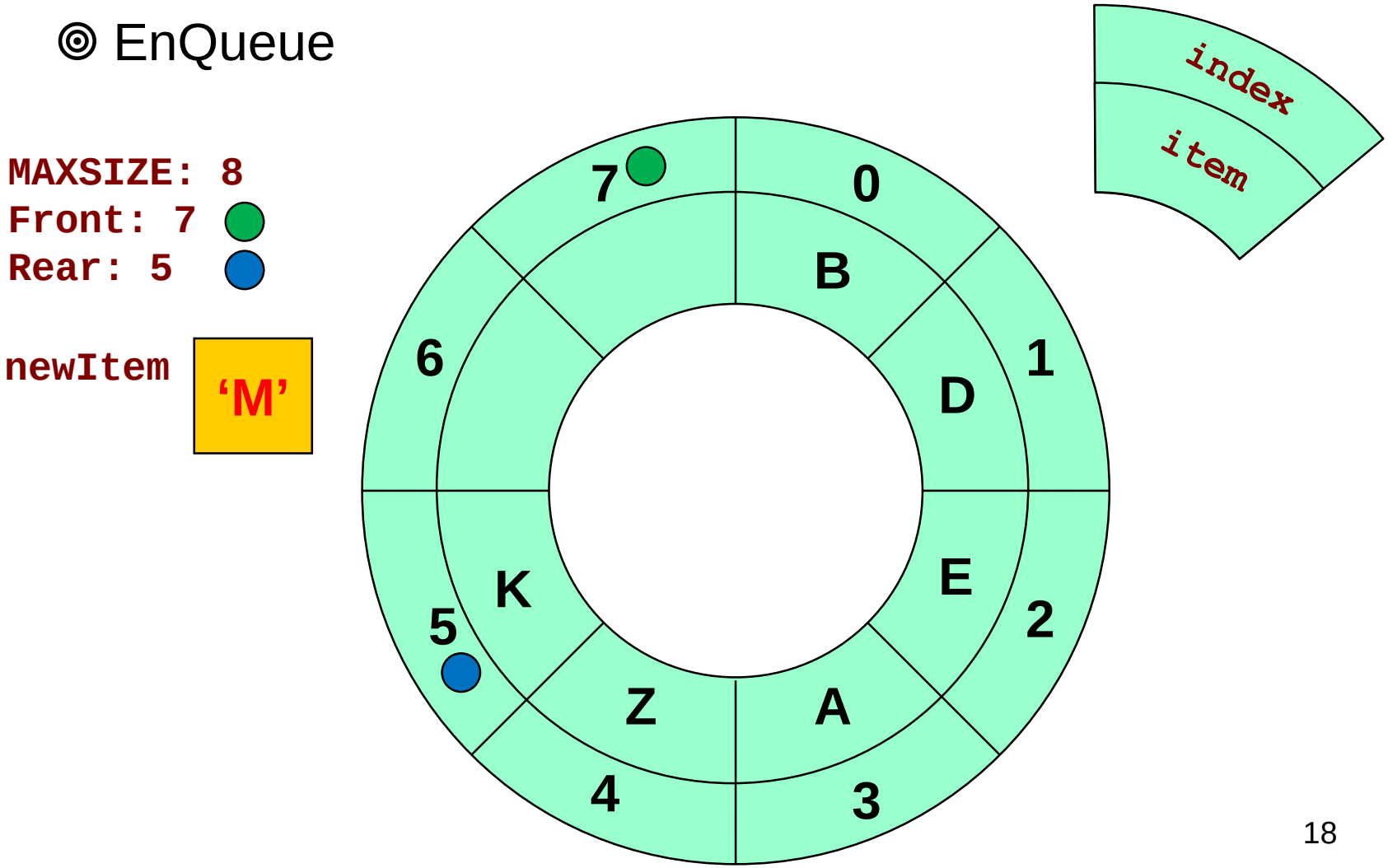
Lab03: Reference (1/3)

◎ EnQueue



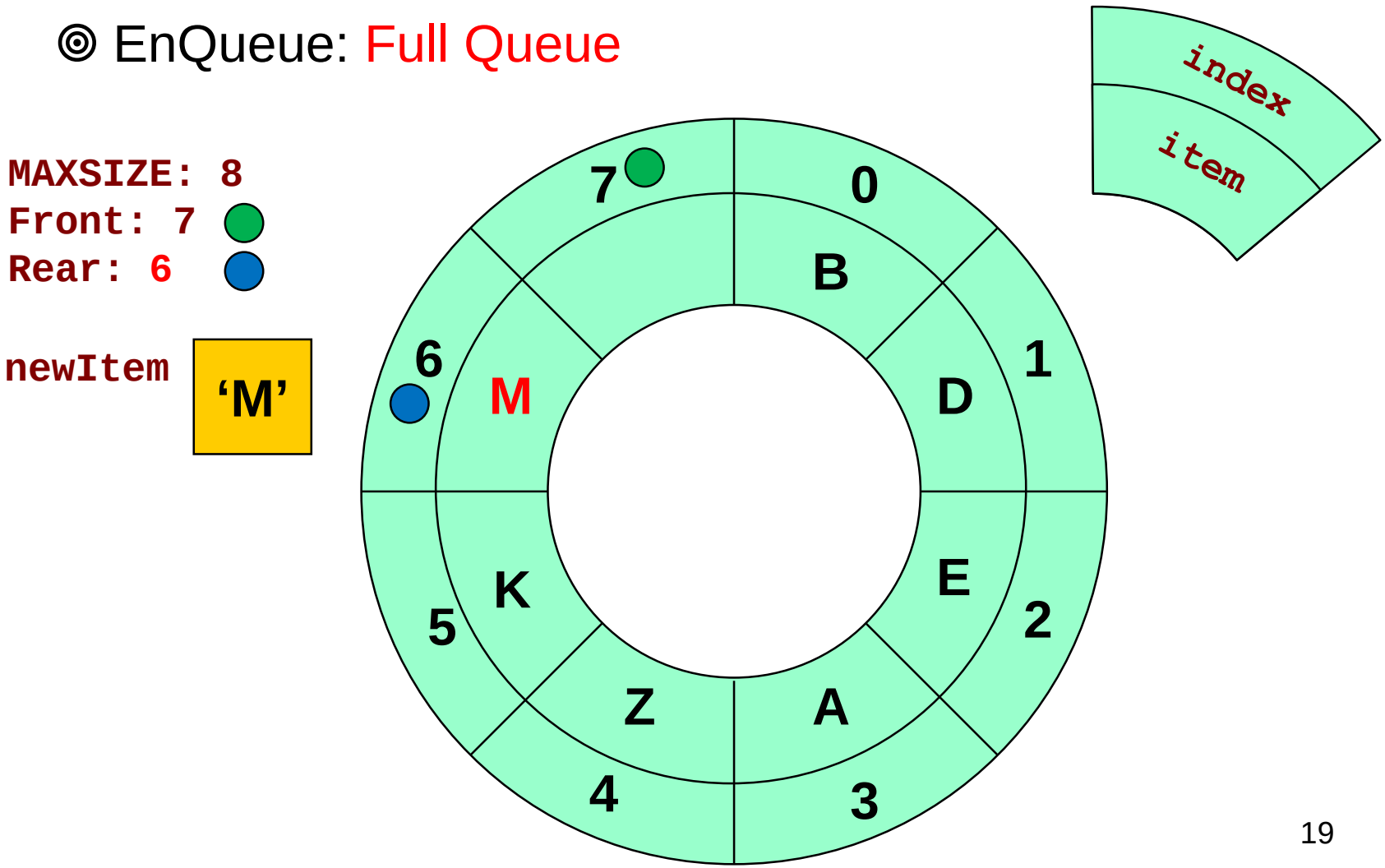
Lab03: Reference (1/3)

◎ EnQueue



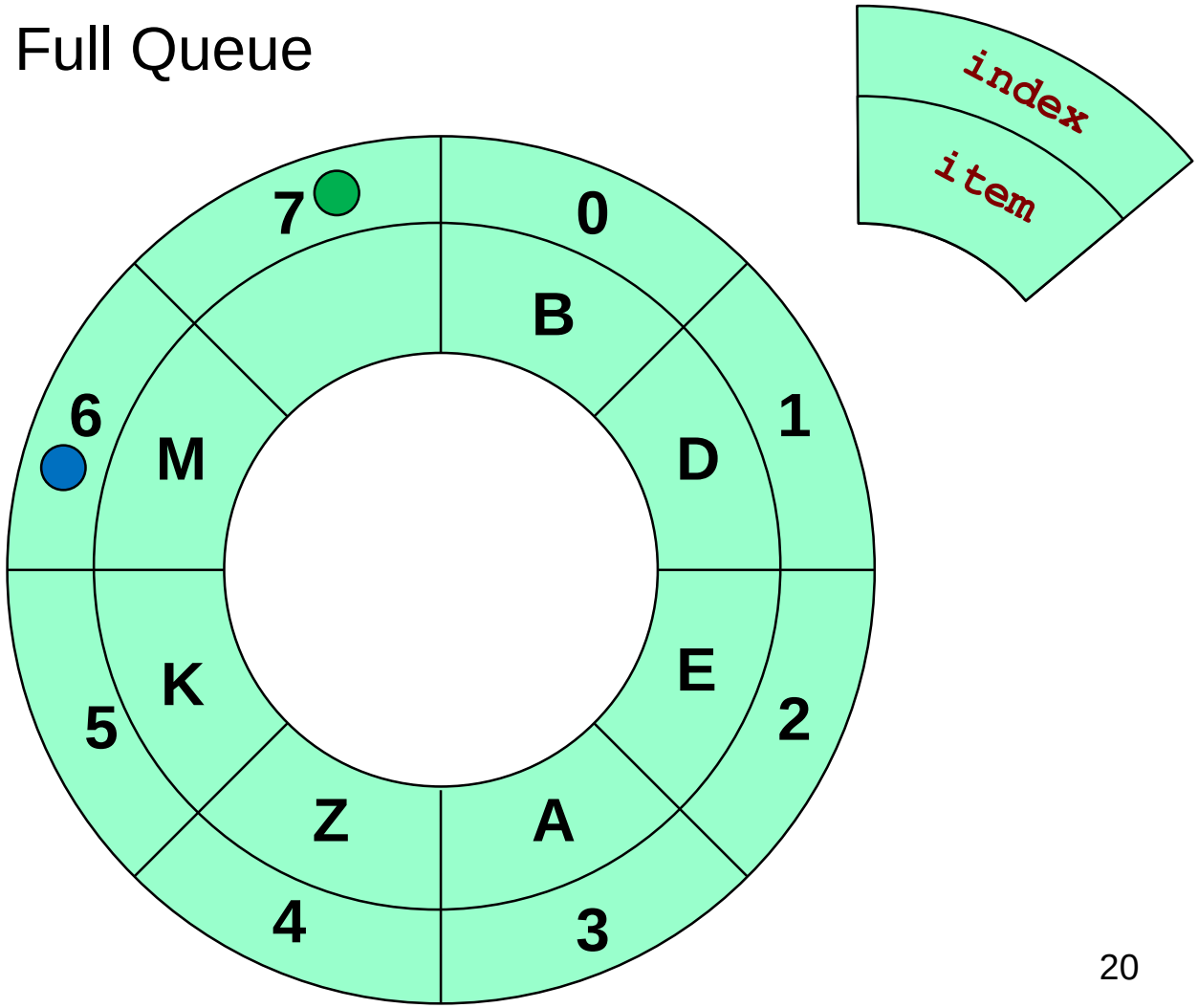
Lab03: Reference (1/3)

◎ EnQueue: Full Queue



Lab03: Reference (2/3)

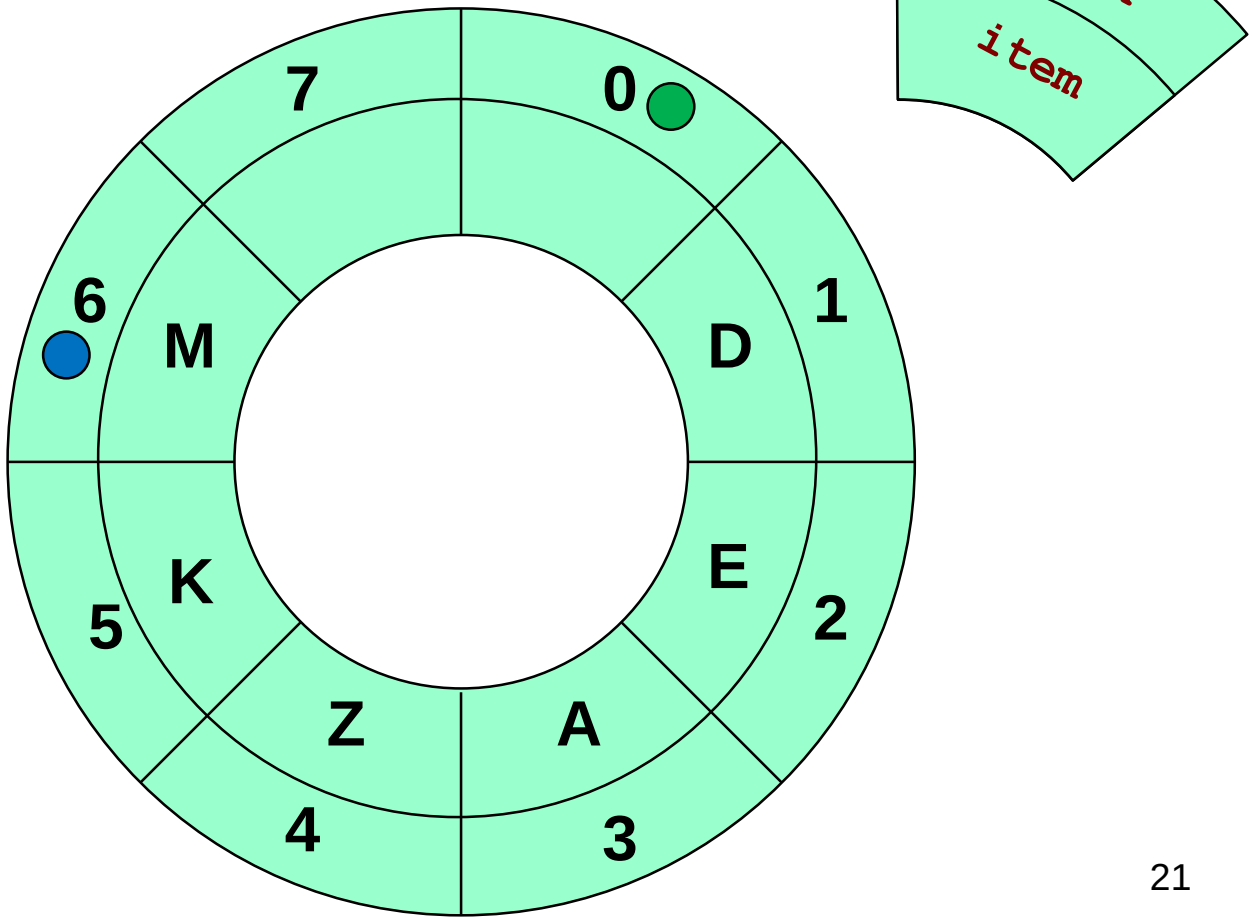
◎ DeQueue: Full Queue



Lab03: Reference (2/3)

◎ DeQueue

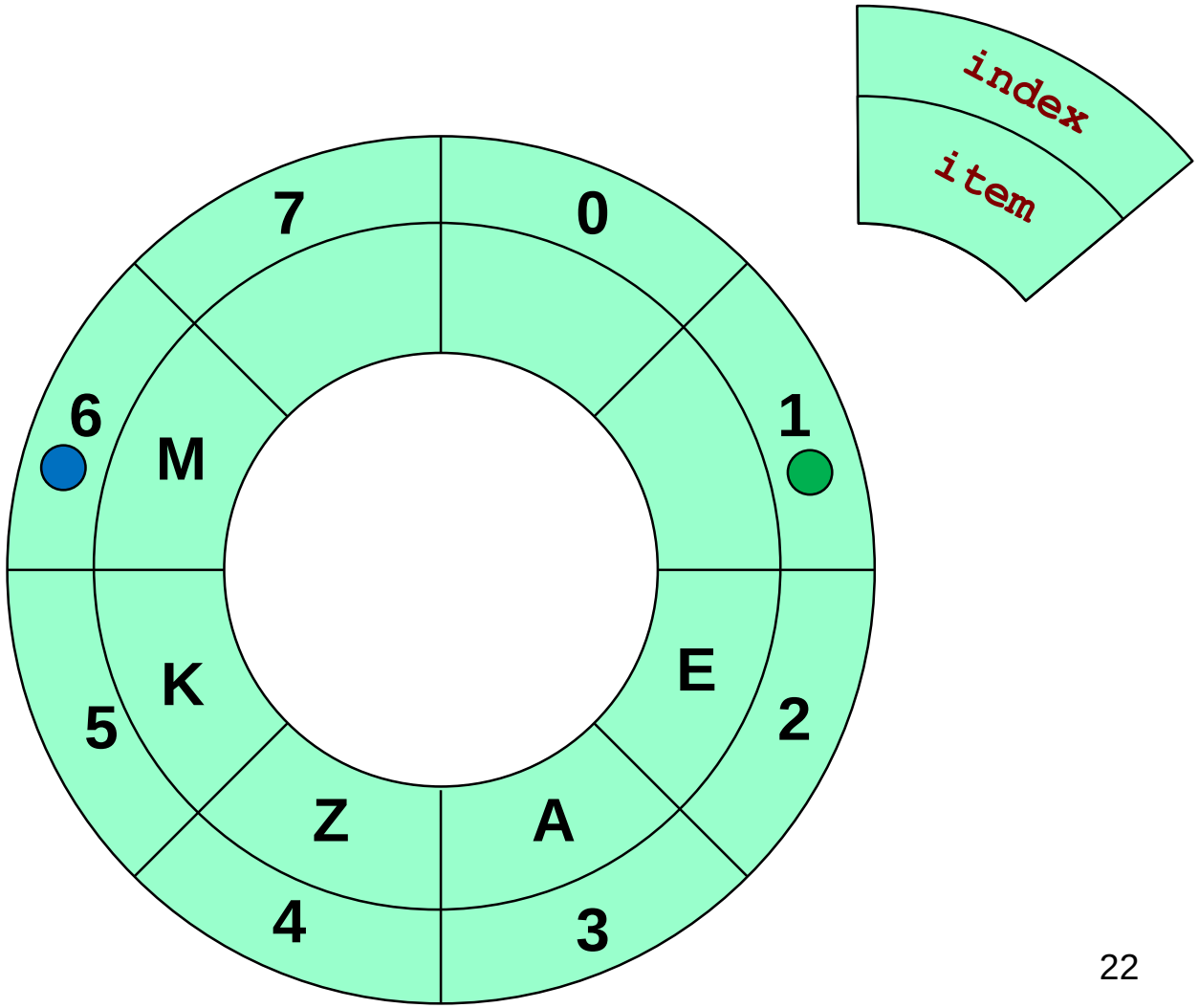
MAXSIZE: 8
Front: 0
Rear: 6



Lab03: Reference (2/3)

◎ DeQueue

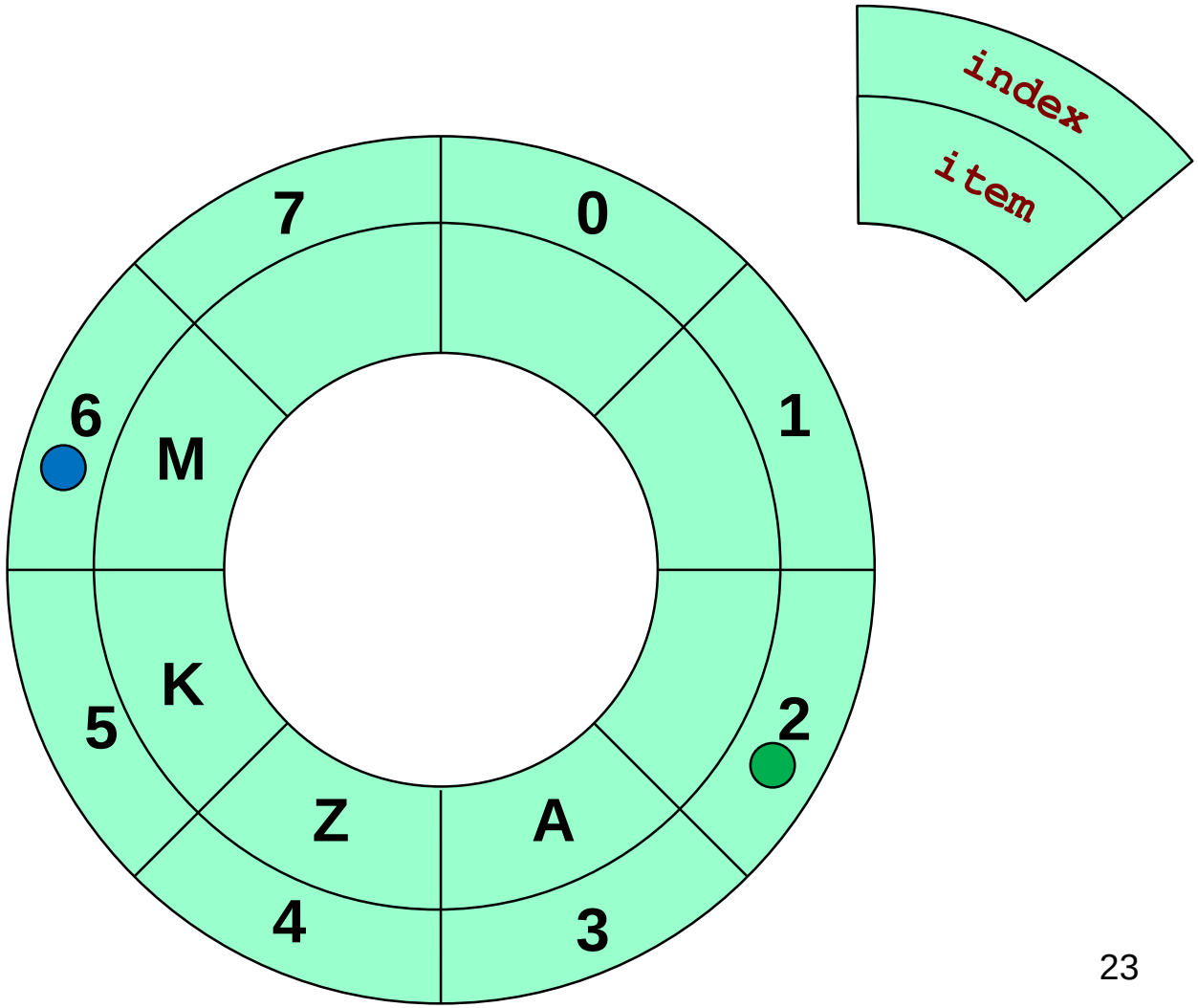
MAXSIZE: 8
Front: 1
Rear: 6



Lab03: Reference (2/3)

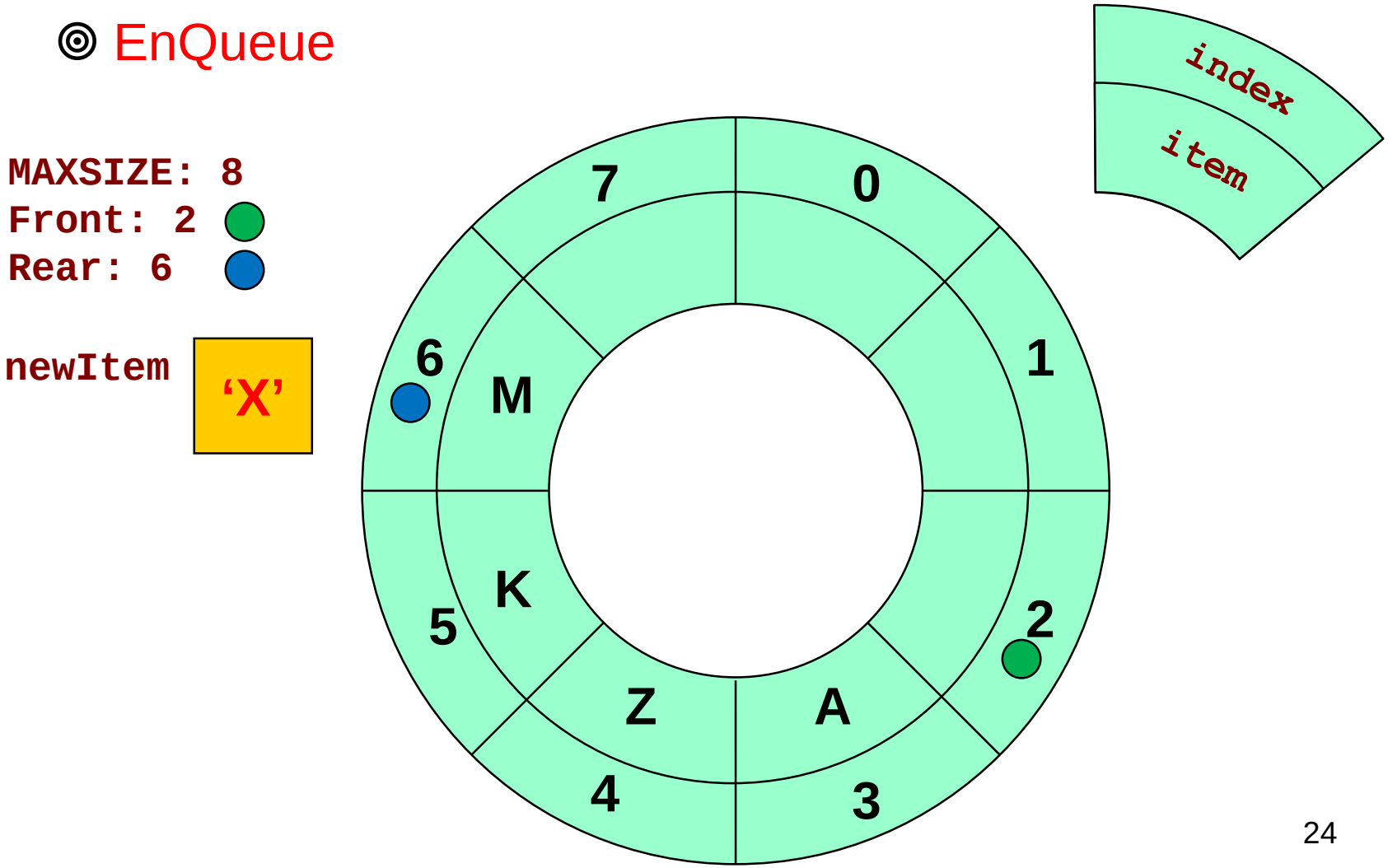
◎ DeQueue

MAXSIZE: 8
Front: 2
Rear: 6



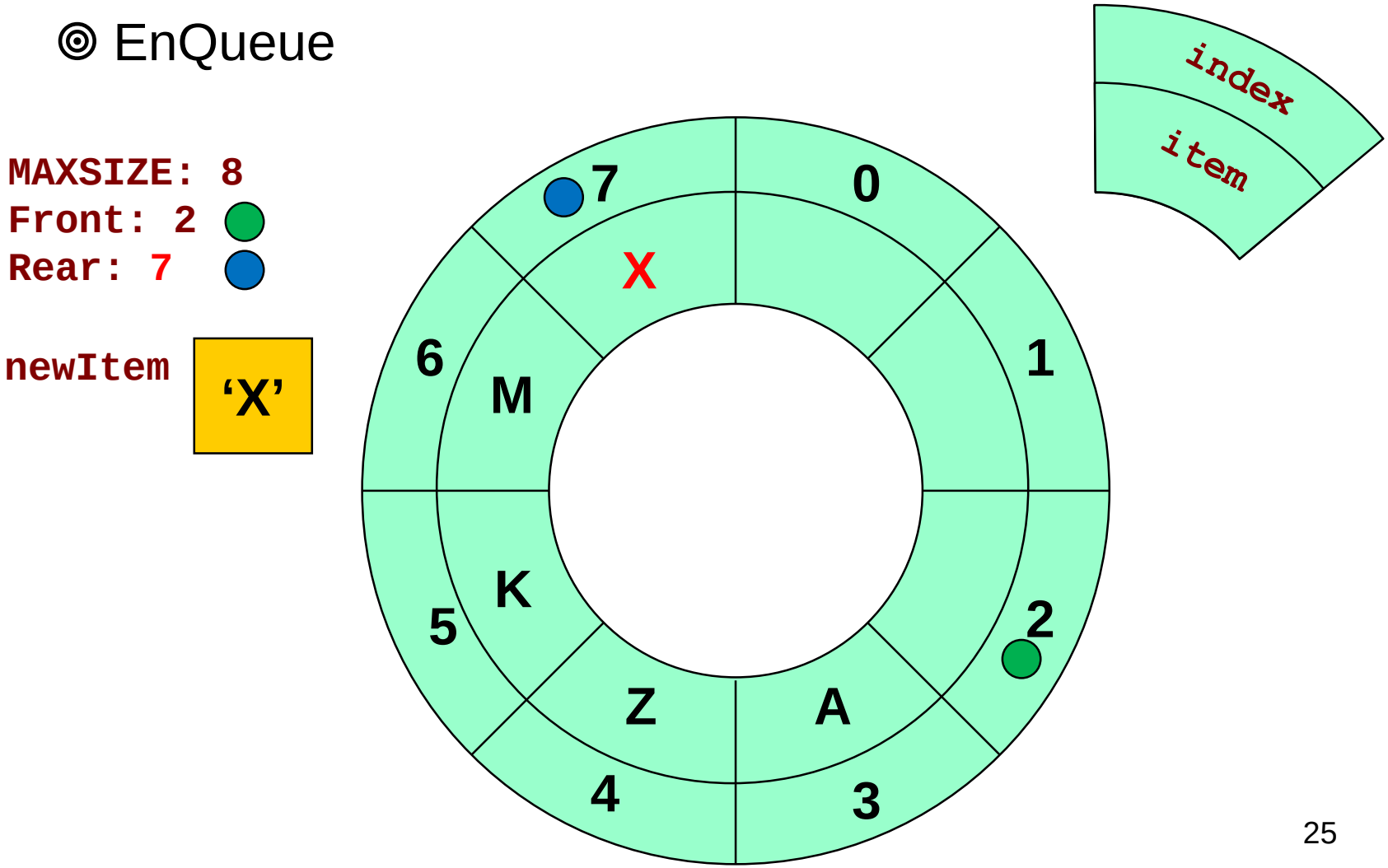
Lab03: Reference (3/3)

- EnQueue



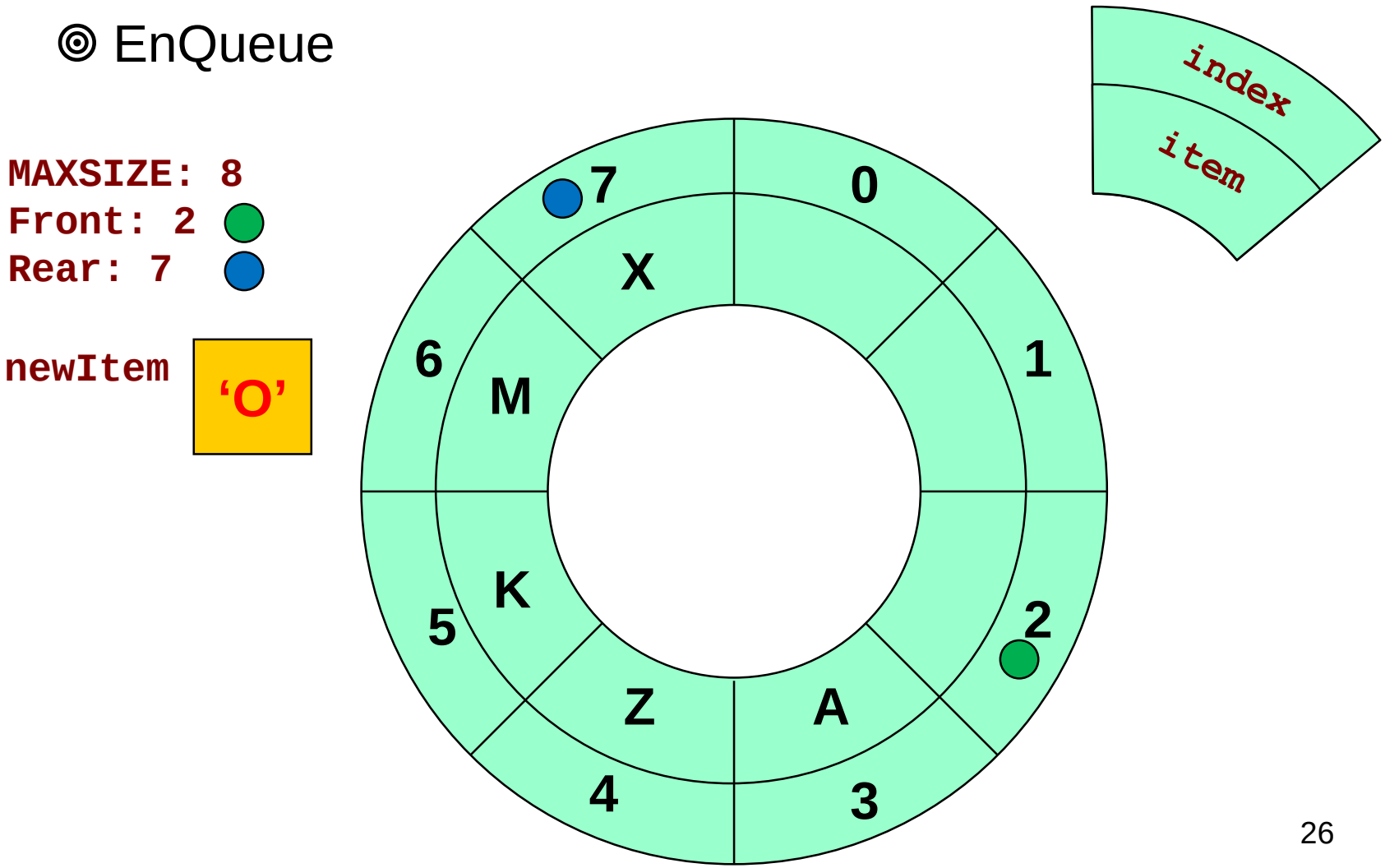
Lab03: Reference (3/3)

◎ EnQueue



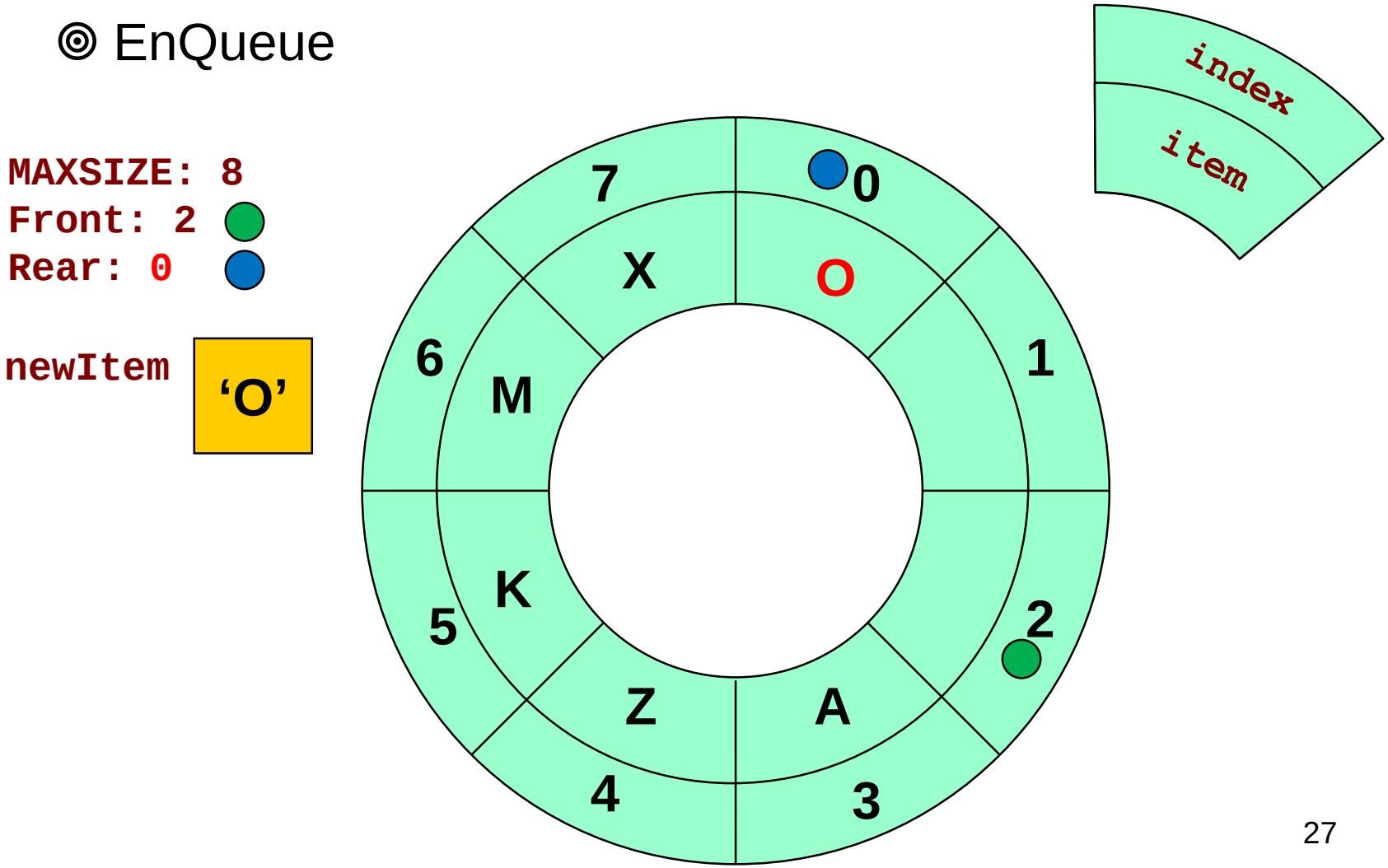
Lab03: Reference (3/3)

◎ EnQueue



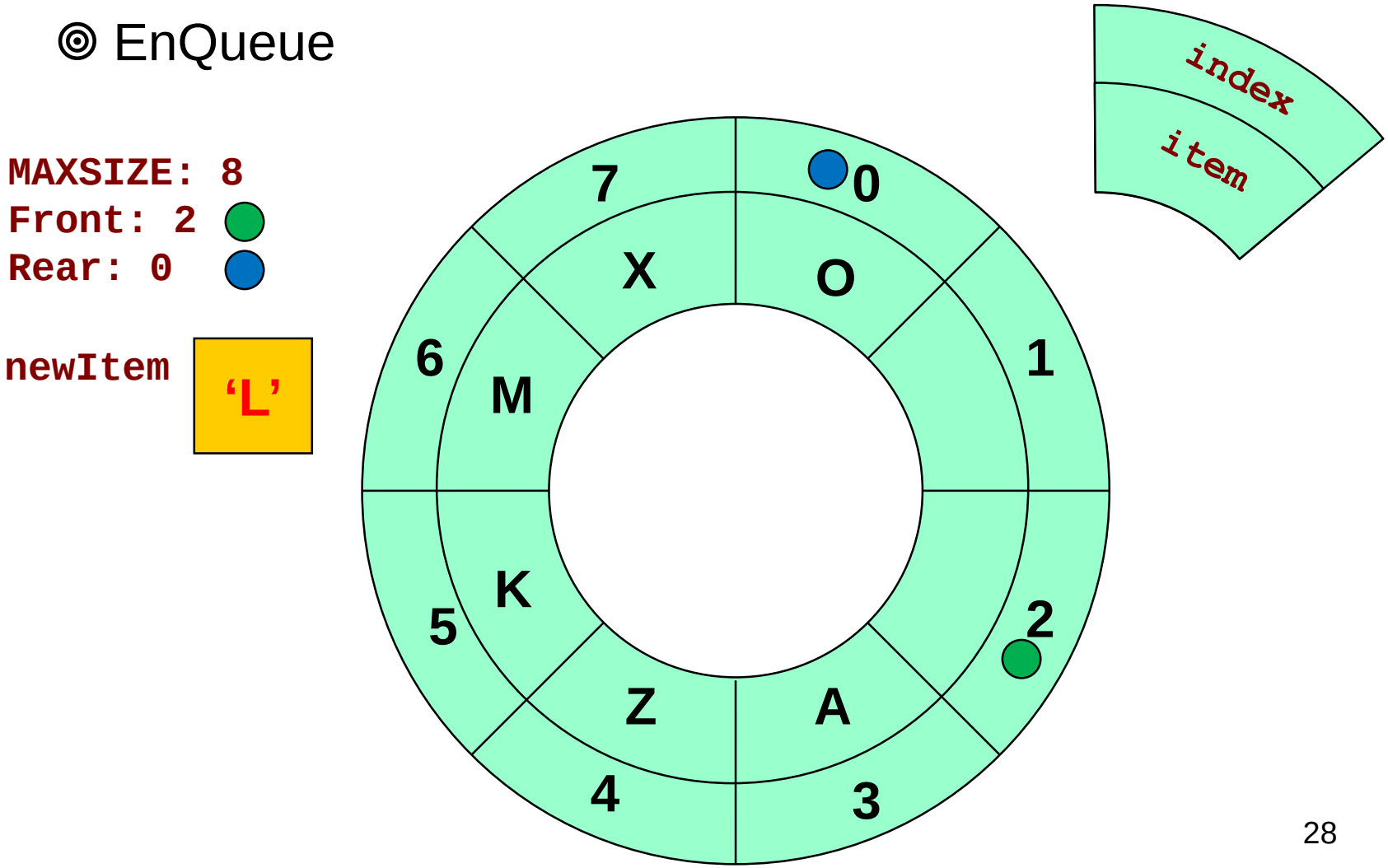
Lab03: Reference (3/3)

◎ EnQueue



Lab03: Reference (3/3)

◎ EnQueue



Lab03: Reference (3/3)

◎ EnQueue: Full Queue