

Data Structures and Algorithms

Mohammad GANJTABESH

*Department of Computer Science,
School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.*

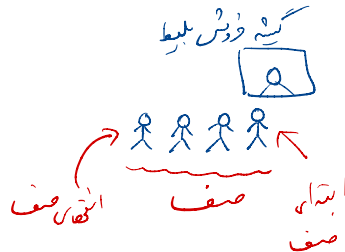
mgtabesh@ut.ac.ir



*Dept. of Computer Science
University of Tehran*

Queue

First In First Out (FIFO)



Data Structures and Algorithms
Undergraduate course



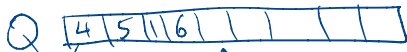
Mohammad GANJTABESH
mgtabesh@ut.ac.ir



Dept. of Computer Science
University of Tehran

Basics

Array



فقط /
عنصر اول
صف

$O(n)$
X

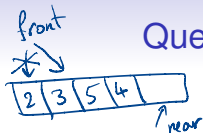
افزودن کردن
به عنصر به انتهای
صف

$O(1)$

Linked List

- بدون هیچ محدودیت
- کارایی بالای عملیات

Queue implementation: Array vs. Linked List



حذف از ابتدای آرایه زمان بر است

ظرفیت محدود دارد (از قبل به پیش مشخص شده)

عملیات در زمان ثابت قابل انجام است
ظرفیت نامحدود

Array

```
class Queue {
    int *Q;
    int capacity;
    int front;
    int rear;
```

```
Queue (int n) {
    Q = new int[n]; capacity = n;
    front = rear = 0;
}
```

3 - }

cache friendly

Linked List

```
class Queue {
    LinkedList L;
    Queue () {
        L = new LinkedList();
    }
    4 methods
}
```

}

Array:

Queue: Enqueue operation

```
void Enqueue (int x) {
```

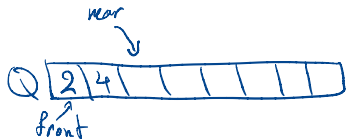
```
    if ((rear + 1) % capacity == front)
        exception ("overflow");
```

```
    Q[rear] = x;
```

```
    rear ++;
```

```
}
```

Enqueue (2);
" (4);



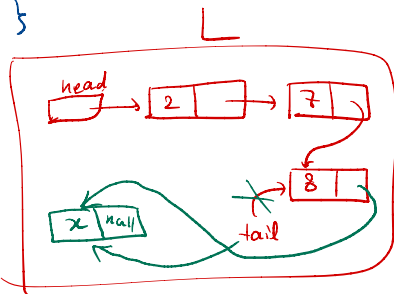
Linked List: ایمانه کردن یک عنصر به انتهای صف

Linked List:

```
void Enqueue (int x) {
```

```
    L.AddLast(x);
```

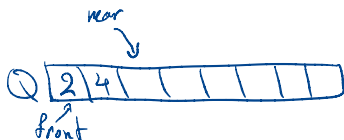
```
}
```



Queue: Dequeue operation

Array

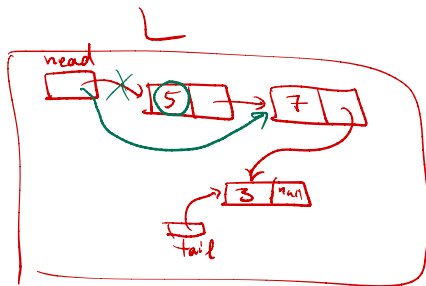
```
int Dequeue () {  
    if (front == rear)  
        exception ("Underflow");  
    int r = Q[front];  
    front = (front + 1) % capacity;  
    return (r);  
}
```



Dequeue()

Linked List

```
int Dequeue () {  
    return (L.RemoveFirst());  
}
```



Applications

- پردازش درخواست های ورودی منابع کم. (پردازنده، سرور، اینترنت)

- پیمایش سطحی گراف. (BFS).

- صف های اولویت (یک طرفه / دو طرفه).