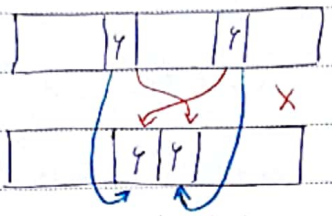
Subject: جلسه ۱۸
Year. ۱۴۰۰ Month. ۹ Date. ۷ ( )

مرتب سازی

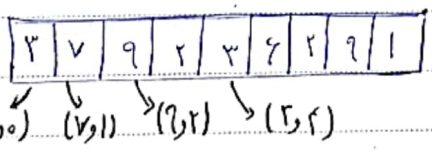
مرتب سازی: آرایه A شامل n عنصر داده شده است. A را مرتب کنید.

* درجا بودن: برای مرتب سازی نیاز به حافظه اضافه نباشد. یا حداکثر $O(1)$
* پایدار بودن: ترتیب عناصر یکسان بعد از مرتب سازی حفظ شود.



مرتب سازی	زمان بدترین	زمان متوسط	درجا (in-place)	پایدار (stable)
مرتب سازی حبابی	$O(n^2)$	$O(n^2)$	✓	✓
مرتب سازی درجی	$O(n^2)$	$O(n^2)$	✓	✓
مرتب سازی انتخابی	$O(n^2)$	$O(n^2)$	✓	X
مرتب سازی ادغامی	$O(n\log n)$	$O(n\log n)$	X (میتونه نباشه)	✓
مرتب سازی هرمی	$O(n\log n)$	$O(n\log n)$	✓	X
مرتب سازی سریع	$O(n^2)$	$O(n\log n)$	X	✓

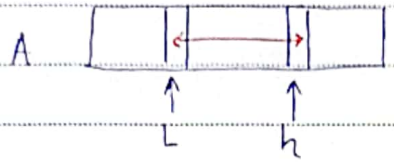
* هر مرتب سازی که پایدار است را می خواهیم پایدار کنیم (با استفاده از حافظه اضافه)



اگر عنصر اول شون کی جود عنصر دوم را مقایسه می کنیم.

مرتب سازی سریع (Quick Sort)

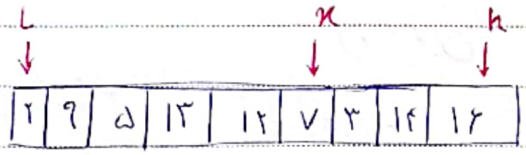
- زیر برنامه Partition(L, h, x) آرایه شامل عناصر حاضر در اندیس های L تا h را در نظر بگیرید.
  $L \le x \le h$



- قرار دادن عنصر A[x] در محل درست آن در حالت مرتب شده

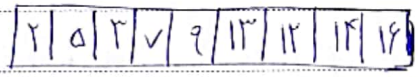
و برگرداندن اندیس آن

- عناصر کوچک تر از A[x] در سمت چپ آن و عناصر بزرگ تر از A[x] در سمت راست آن قرار گیرد.



حل Partition را می توانیم در زمان O(n) انجام داد (n تعداد عناصر)

```
Partition(L,h){
i=L
Pivot= A[h]
for(j: L → h-1)
if (A[j] ≤ Pivot)
Swap(A[i], A[j])
i++
}
Swap(A[i], A[j])
return j
```




```
Partition(L,h,x){
Swap(A[x], A[h])
Partition(L,h)
}
```

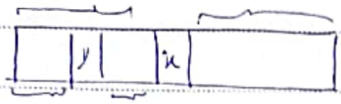
Subject: جلسه ۱۸
Year. ۱۴۰۰ Month. ۹ Date. ✓ ( )

مرتب سازی سریع

[L,h]

۱. آرایه را به نحوی عنصر دلخواه x را Partition کن. (فرض کنید P حاصل x باشد)

۲. دو زیر آرایه [۰,P-۱] و [P+۱,h] را به طور بازگشتی و متوسط همین الگوریتم QS مرتب کنید.




```
QuickSort (L,h) {
    if (L>h)
        return
    x one element in [L,h]
    P = Partion(L,h,x)
    QuicSort (L,P-1)
    QuicSort (P+1,h)
}
```

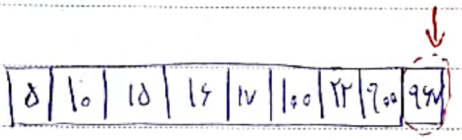
$x = \frac{L+h}{۲}$ $O(n^۲)$

$x = L$ $O(n^۲)$

$x = h$ $O(n^۲)$

$x = random(L,h)$

	پایدار	درجا	متوسط	زمان در بدترین حالت	
Partition ← مبنای	✓	✓	$O(n \log n)$	$O(n^۲)$	Quick Sort



$T(n) = T(n-۱) + O(n) = O(n^۲)$