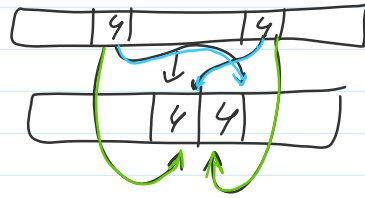


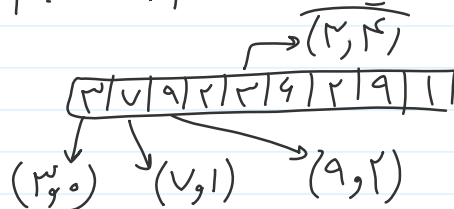
موضوع: مرتب سازی - 1

- مرتب سازی آرایه A شامل n عنصر داده شده است. A را مرتب کنید.
- * درجا بودن: برای مرتب سازی نیاز به حافظه اضافی نباشد. یا حداقل $O(1)$
- * پایدار بودن: ترتیب عناصر یکسان بعد از مرتب سازی حفظی شود.



Stable پایدار	in-place درجا	زمان (متوسط)	زمان (بدترین)	مرتب سازی صبابی
✓	✓	$O(n^2)$	$O(n^2)$	
✓	✓	$O(n^2)$	$O(n^2)$	مرتب سازی ایزرگی
X	✓	$O(n^2)$	$O(n^2)$	مرتب سازی انتخابی
✓	X (نی ترینه باشه)	$O(n \log n)$	$O(n \log n)$	مرتب سازی ادغامی
X	✓	$O(n \log n)$	$O(n \log n)$	مرتب سازی هری
✓	X	$O(n \log n)$	$O(n^2)$	مرتب سازی درجای

- * همتب سازی که پایدار نیست رای توانیم پایدار کنیم (با استفاده از حافظه اضافی)

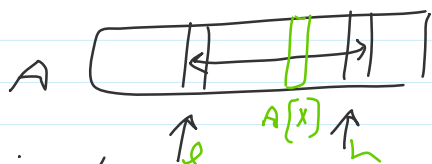


مرتب سازی سریع (quick-sort) (choare) (1959)

- * زیرمساله: $Partition(l, h, x)$: آرایه شامل عناصر حاضر در اندیس l تا h را بر اساس عنصر x مرتب کند.



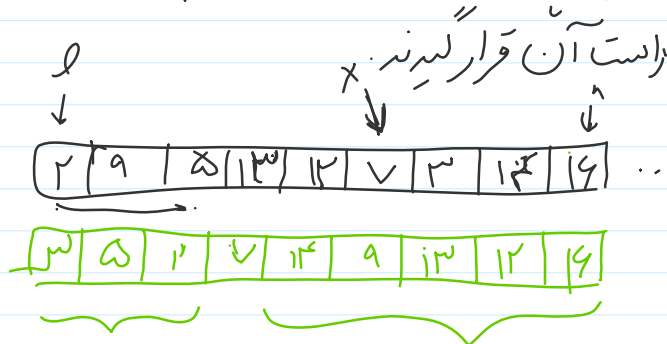
* زیرمساله: $Partition(l, h, x)$: از این سائل عناصر حاصل اندیس l



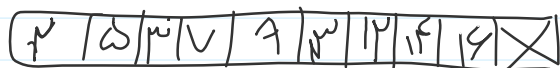
$l \leq h$ را در نظر بگیرد هدف:

* قرار دادن عنصر $A[x]$ در محل درستش در حالت مرتب شده و برگرداندن اندیس آن

* عناصر کوچکتر از $A[x]$ درست چپ آن و عناصر بزرگتر از $A[x]$ درست



* عمل $Partition$ را می توان در $O(n)$ انجام داد. (تعداد عناصر n)



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

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

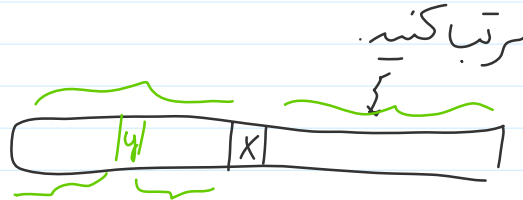
```

return(h)
    }
    swap(A[i], A[j])
    }

```

مرتب‌سازی سریع
[l, h]

- ۱- آرایه را بر حسب عنصر دلخواه x پارتیشن کن [فرض کن P محل x باشد]
- ۲- دو زیر آرایه $A[l, P-1]$ و $A[P+1, h]$ را به طور بازگشتی مرتب کن همین الگوریتم



quick sort(l, h) {

if (l > h)
return

x: one element in [l, h]

→ Partition(l, h, x)
quick sort(l, P-1)
quick sort(P+1, h)

$n = l$

$n = h$

$n = \frac{l+h}{2}$

$n = \text{random}(l, h)$

پایه

درجا

زمان متوسط

$O(n \log n)$

زمان بدترین

$O(n^2)$

quick sort

(مستقیماً به پارتیشن)
دلدار

$T(n) = T(n-1) + O(n)$

$$[2 | 1. | 15 | 19 | 10 | 1.. | 22 | 9.. | 991] \rightarrow I(n) = I(n-1) \cup I(1) \\ = O(n^2)$$

