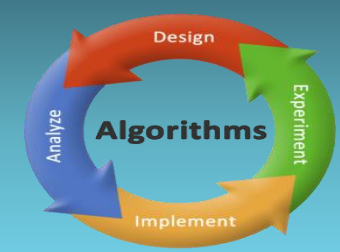
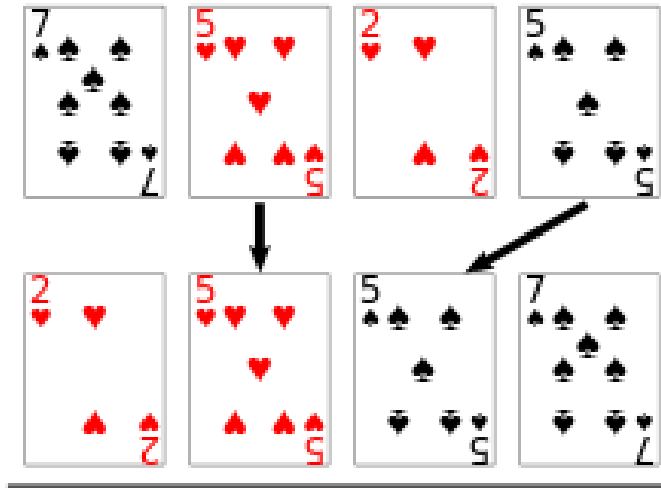


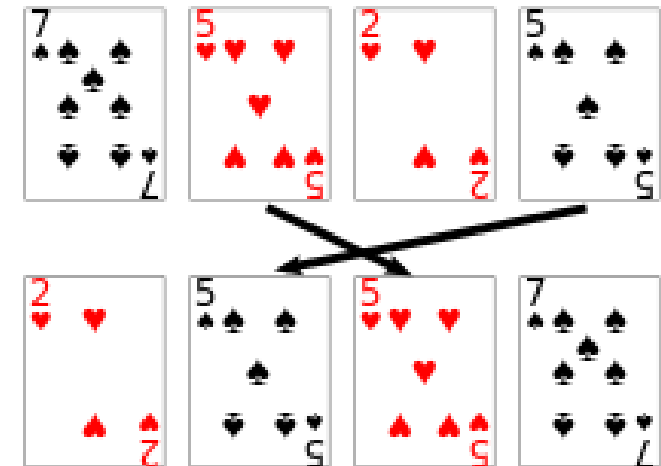
مرتب سازی

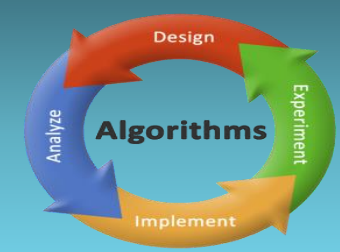


Stable



Not stable



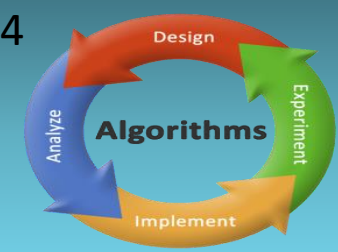


# مرتب‌سازی سریع (Quick Sort)

این الگوریتم همانند مرتب‌سازی ادغامی از روش جداسازی و حل (Divide and Conquer) استفاده می‌کند. در هر مرحله یک عنصر را در سر جای خود قرار می‌دهد و آرایه را به دو قسمت تقسیم می‌کند و دوباره الگوریتم را برای آرایه‌های جدید فراخوانی می‌کند. □

QUICKSORT( $A, p, r$ )

```
1  if  $p < r$ 
2       $q = \text{PARTITION}(A, p, r)$ 
3      QUICKSORT( $A, p, q - 1$ )
4      QUICKSORT( $A, q + 1, r$ )
```



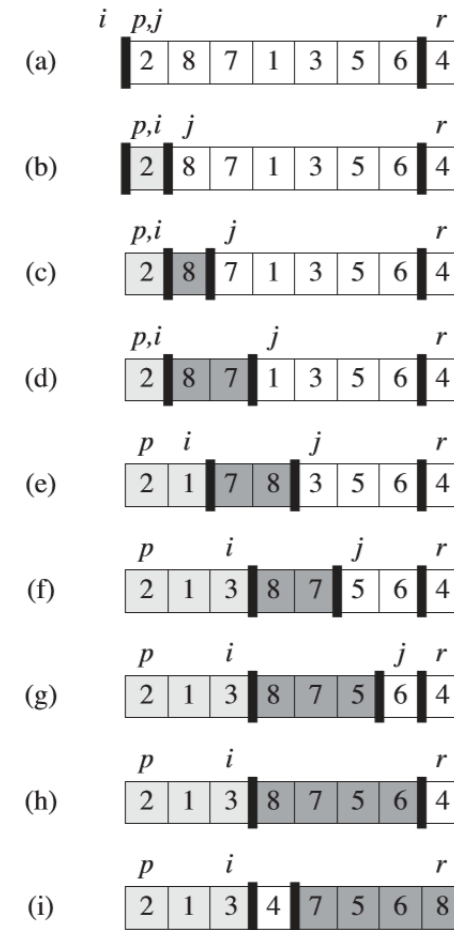
# Quick sort

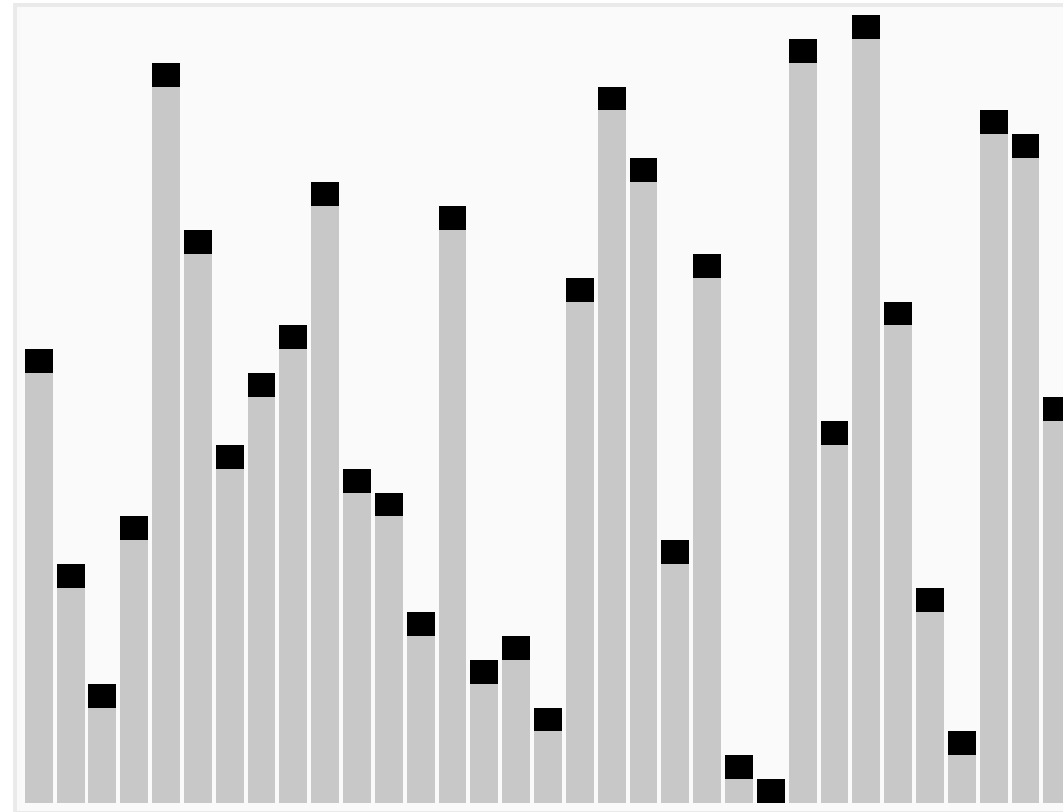
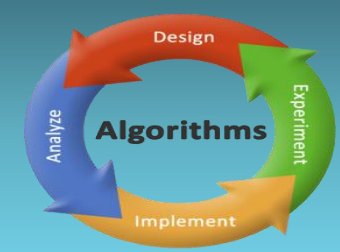
PARTITION( $A, p, r$ )

```

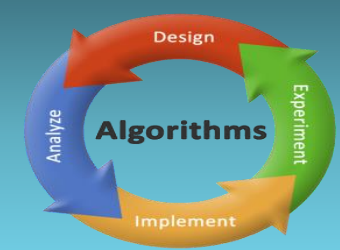
1   $x = A[r]$ 
2   $i = p - 1$ 
3  for  $j = p$  to  $r - 1$ 
4      if  $A[j] \leq x$ 
5           $i = i + 1$ 
6          exchange  $A[i]$  with  $A[j]$ 
7  exchange  $A[i + 1]$  with  $A[r]$ 
8  return  $i + 1$ 

```





انیمیشن مرتب سازی سریع

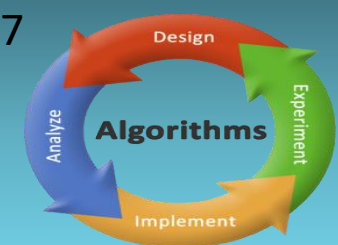


# زمان اجرا

- ❑ به راحتی می توان فهمید که الگوریتم قسمت بندی (Partition) دارای  $\Theta(n)$  است.
- ❑ در مورد Quick Sort چطور؟
- ❑ رابطه بازگشتی زمان را برای quick sort را چطور نوشت؟
- ✓ این رابطه وابسته به چگونگی تقسیم شدن مسائل است.

QUICKSORT( $A, p, r$ )

```
1  if  $p < r$ 
2       $q = \text{PARTITION}(A, p, r)$ 
3      QUICKSORT( $A, p, q - 1$ )
4      QUICKSORT( $A, q + 1, r$ )
```



# بررسی زمان اجرای مرتب سازی سریع (Quick Sort)

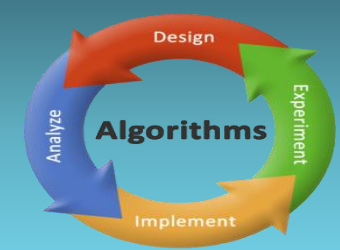
❑ حال باید بررسی کنیم که بدترین زمان اجرای الگوریتم مرتب سازی سریع مربوط به چه موقع است؟  
 ✓ بدترین حالت زمانی رخ می دهد که در مرحله ی تقسیم، داده ها آن طور که باید تقسیم نشوند؛ یعنی آرایه های حاصل دارای طول های 1 و  $n-1$  باشند:

$$T(n) = T(n - 1) + \Theta(n) \Rightarrow T(n) = O(n^2)$$

❑ بهترین حالت چطور؟  
 ✓ زمانی هست که آرایه به دو قسمت مساوی تقسیم شود یعنی:

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) \Rightarrow T(n) = O(n \lg n)$$

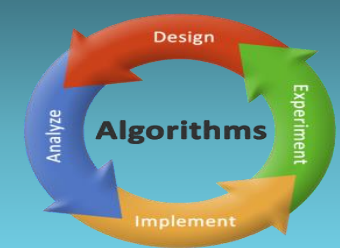
❑ دارای بدترین زمان اجرای  $O(n^2)$  و زمان میانگین  $O(n \lg n)$  می باشد.



## آنچه آموختیم در یک نگاه

Name	Best case	Avg case	Worst case	Stable
Insertion sort	$n$	$n^2$	$n^2$	yes
Merge sort	$n \log n$	$n \log n$	$n \log n$	yes
Heap sort	$n \log n$	$n \log n$	$n \log n$	no
Quick sort	$n \log n$	$n \log n$	$n^2$	no





# پایداری (stability)

یکی از ویژگی‌هایی که می‌توان در مورد الگوریتم‌های مرتب‌سازی بررسی کرد، پایداری است؛ یعنی عناصر با مقدار یکسان ترتیب خود در ورودی را، در خروجی نیز حفظ کنند. □

