

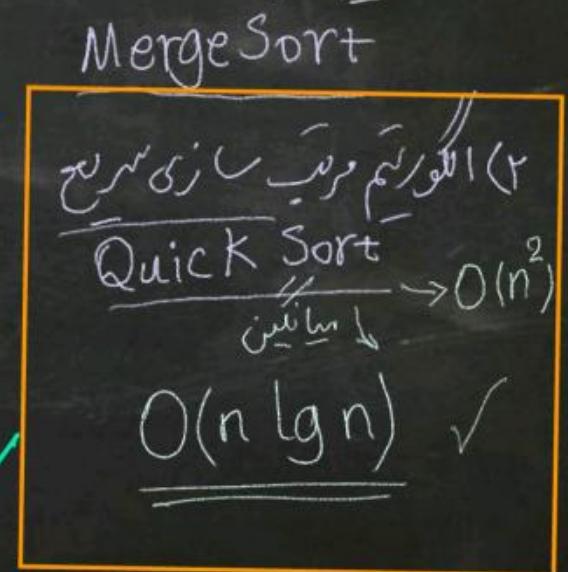
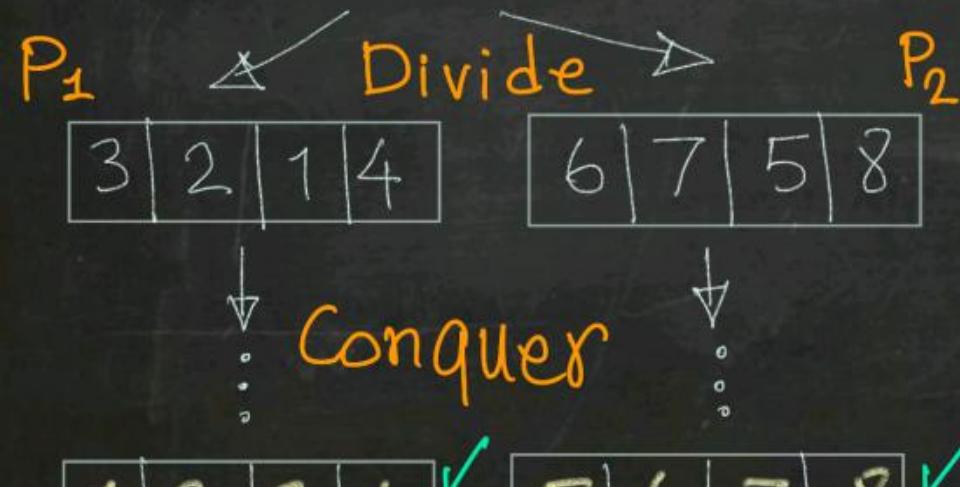
مرتب سازی ب اوسن فرم و غلبہ - مرتب سازی سرع

Problem P

6	3	7	2	5	8	1	4
---	---	---	---	---	---	---	---

Sorting using
Divide and Conquer

الgoritم مرتب سازی اعجمی



نگرش فرم و غلب بر طراحی الگوریتم

1) مسئلہ اصل را P بنایا. مسئلہ P را به تعدادی زیر مسئلہ

✓ Divide

P_1, P_2, \dots, P_K

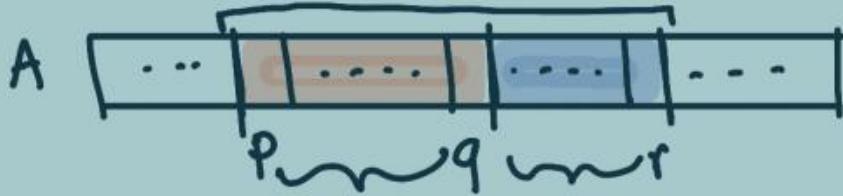
2) هر کدام از زیر مسئلہ های P_i را اگر قدر کافی ساده باشد روشن

حل کند و اگر هموزن قدر کافی ساده نباشد اول وق را بطور زیر کشی

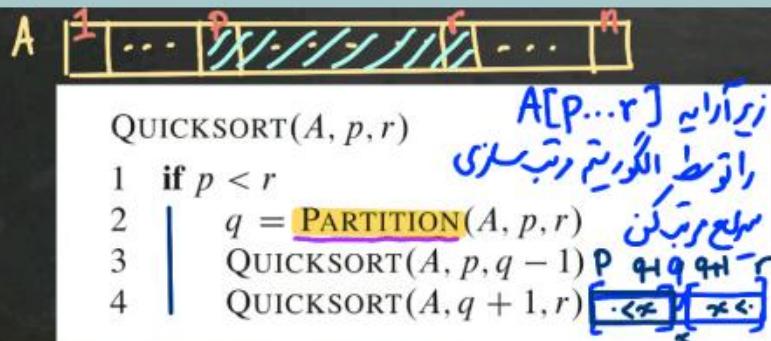
✓ Conquer

3) حواب مسئلہ P را با کسب حواب زیر مسئلہ های P_1, \dots, P_K بدست آوری

✓ Combine



$\forall i, j, p \leq i \leq q < j \leq r, A[i] < A[j]$



Pivot
عنصر محوری
 x

مرتب سازی سعی
Quick Sort

با استفاده از ایده لغزشی و مکالمه

افزار

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** $i + 1$

in-place, نه برای این ایندکس هایی که در اینجا نشان داده شده اند
برای مرتب سازی بسیار اوران

نمایش افزاں

نمایش افزاں . مرتب سازی بازگشتی کردنی دو زیرآرایه کل کریم کو .

نمایش افزاں . ندارد .

PartitioningVerify(A, P, q, r)

for i = P to q

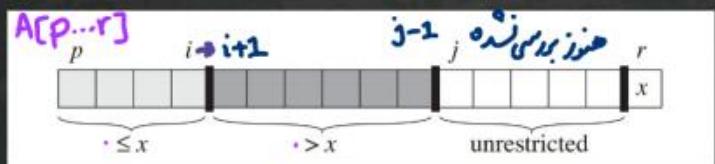
 for j = q+1 to r

 if A[i] > A[j] then return False

return True

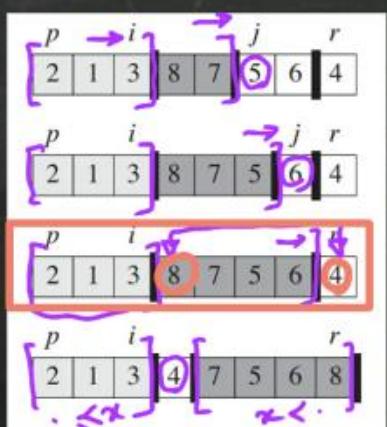
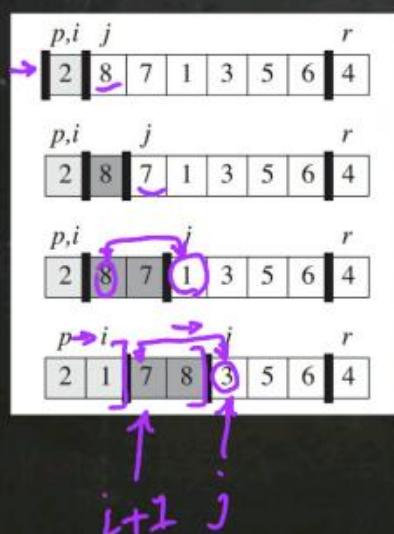
اللُّوْرِيْمَ افْرَاز

هدف از الُّوْرِيْمَ افْرَاز، جایگا کردن عناصر آرایه A به صورت که سُرطانی مفعَّل شود:

$$\forall i < q, \forall j > q, A[i] < A[j]$$


برای الُّوْرِيْمَ افْرَاز ایدهٔ فکری معاونی وجود دارد:

- ✓ Hoare الُّوْرِيْمَ
- ✓ CLRS الُّوْرِيْمَ



PARTITION(A, p, r)

```

1    $x = A[r]$  جزء پسر راست
2    $i = p - 1$  من وحش
3   for  $j = p$  to  $r - 1$  جزء پسر را
4     if  $A[j] \leq x$  ناصیح کنم
5        $i = i + 1$ 
6       exchange  $A[i]$  with  $A[j]$ 
7   exchange  $A[i + 1]$  with  $A[r]$ 
8   return  $i + 1$ 
↓ q

```

آنالیز زمان الگوریتم اوزان

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  $i + 1$ 

```

$$n = r - p + 1$$

: طول زیرگردایی که الگوریتم

(فرز بروی) انجام می‌شود

→ Best case : $\Theta(n)$

→ Worst case : $\Theta(n)$

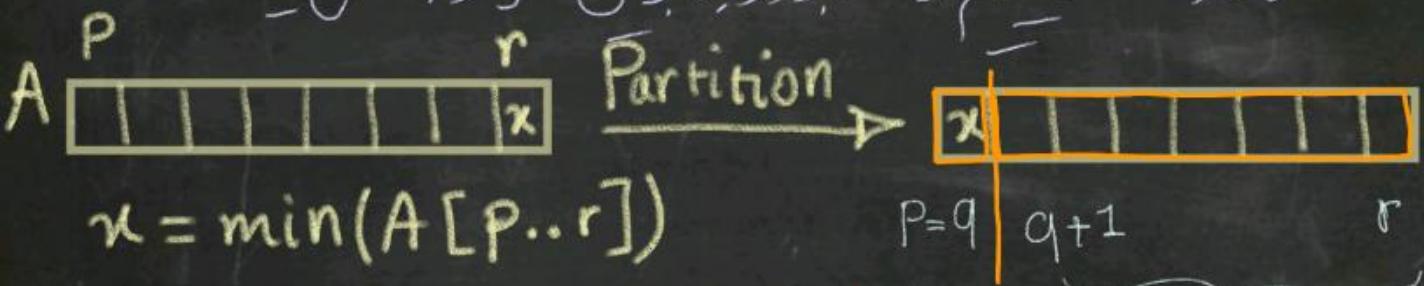
⇒ Every case : $\Theta(n)$

Worst case analysis

آنالیز مرتب سازی سریع - بدترین حالت

- فرض کنید عناصر آرایه ورودی از اندکا به کسر نزولی مرتب شده‌اند.

آنلاین مختصر تحریر، بعد از مسح خواهد بود و سایر اجزای سطح زیر است:



QuickSort($A, p, q-1$) $\rightarrow T(0) = O(1)$
 QuickSort($A, q+1, r$) $\rightarrow T(n-1)$

نامناسب تر از این ریسب لوچ خواهد بود:

$$\underline{n := r - p + 1}$$

آنالیز مرتب سازی سرعی - پرینٹ حالت

QuickSort($A, P, q-1$) $\rightarrow T(0) = O(1)$

QuickSort($A, q+1, r$) $\rightarrow T(n-1)$

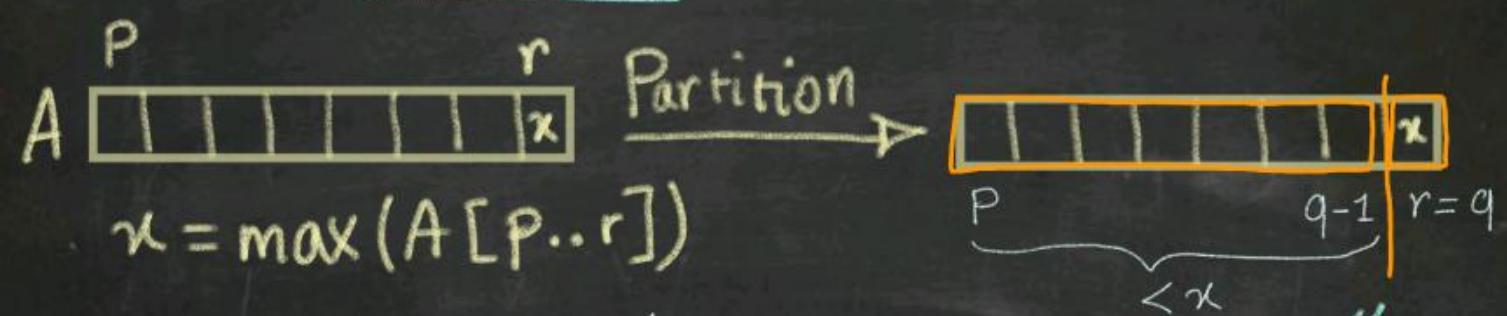
$T(n) = T(n-1) + \cancel{T(0)} + \underbrace{\Theta(n)}_{O(1)}$ بنا بر این داریم
هزینه اجرای افزار

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

$\Rightarrow T(n) = \Theta(n^2)$

آنالیز مرتب سازی سرعی - بدرین حالت

- پیش رسم به اگر آرایه ورودی را باشد هر سُن صنعتی



- ناحیه بزرگتر از κ بدهن تیپ پوچ خواهد بود.

$$T(n) = T(n-1) + T(0) + \Theta(n)$$

$$\Rightarrow T(n) = \Theta(n^2)$$

* هر حالت از این

مرتب سُن صنعتی بدرین حالت الگوریتم مرتب سازی سرعی است.

$$T(n) = T(n-1) + \Theta(n)$$

$$\exists c_1 > 0, \exists n_0, \forall n > n_0$$

$$T(n) \leq T(n-1) + c_1 n$$

$$\exists c_2 > 0, \exists n_0, \forall n > n_0$$

$$T(n) \geq T(n-1) + c_2 n$$

$$\begin{aligned} T(n) &= 2 T(n/2) + \Theta(n) \\ \Rightarrow T(n) &= \Theta(n \lg n) \end{aligned}$$

آمالیز مرتب سازی سریع - ابیرین حالت

- ابیرین حالت برای مرتب سازی سریع آرایه مرتب شده نیست

- آرایه ای را فرض کنید به طور تصوری که هر بار اجرای الگوریتم فراز آرایه

به دو زیر آرایه برابر تقسیم می شوند:

$$T(n) = 2 T(n/2) + \Theta(n)$$

- قبل در خلیل مرتب سازی ادامه ایجاد کردیم:

$$\Rightarrow T(n) = \Theta(n \lg n)$$

کمینه کردن ابیرین حالت را برای $n=19$ ارائه دهد.

آن لیز مرتب سازی سریع - مرتبه میانگین

این تجھیسو QuickSort بروی وردی تصادفی طور میانگین زمان اجرای $\Theta(n \lg n)$ صرف می‌کند.

- کمین (لیل میانگین) (جیسو) برای گزین از بدترین حالت، محدوده ارائه وردی را قبیل از اجرا، طور تصادفی پُر ببرید!

آن لِز مرتب سازی سریع - مرتبه میانگین

که روش جایگزین این است که بجای بر زدن صادر آرایه، صادر تغیری را در

هر بار خوانی

الgoritم افزایش صاف

باشد موقت صاف

در آرایه جایگزین

RANDOMIZED-QUICKSORT(A, p, r)

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

RANDOMIZED-PARTITION(A, p, r)

- 1 $i = \text{RANDOM}(p, r)$ $\alpha = A[i]$
- 2 exchange $A[r]$ with $A[i]$ pivot
- 3 **return** PARTITION(A, p, r)