بسم الله الرحمن الرحیم

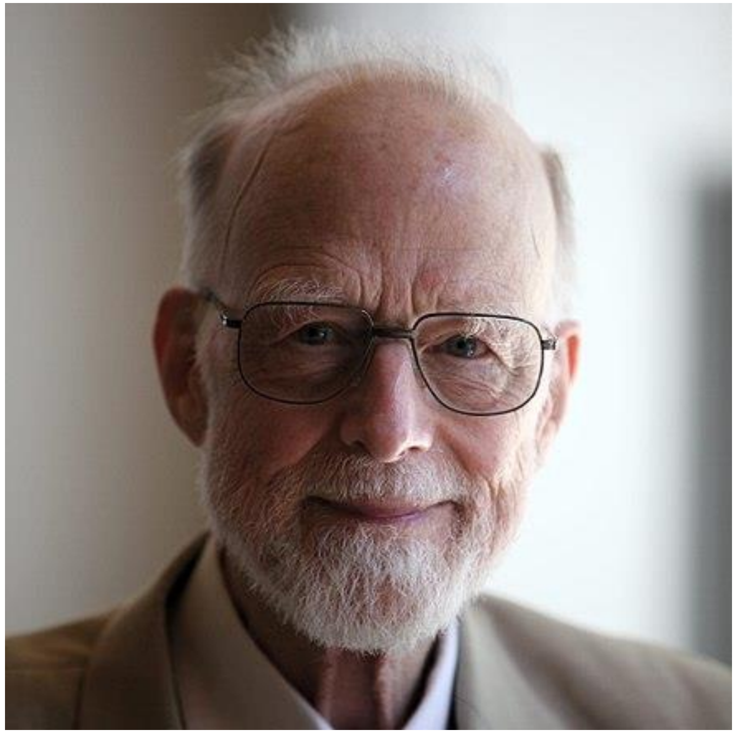
طراحی الگوریتم

(تقسیم و غلبه تصادفی)

دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان

بهار ۱۴۰۰

مرتب‌سازی سریع

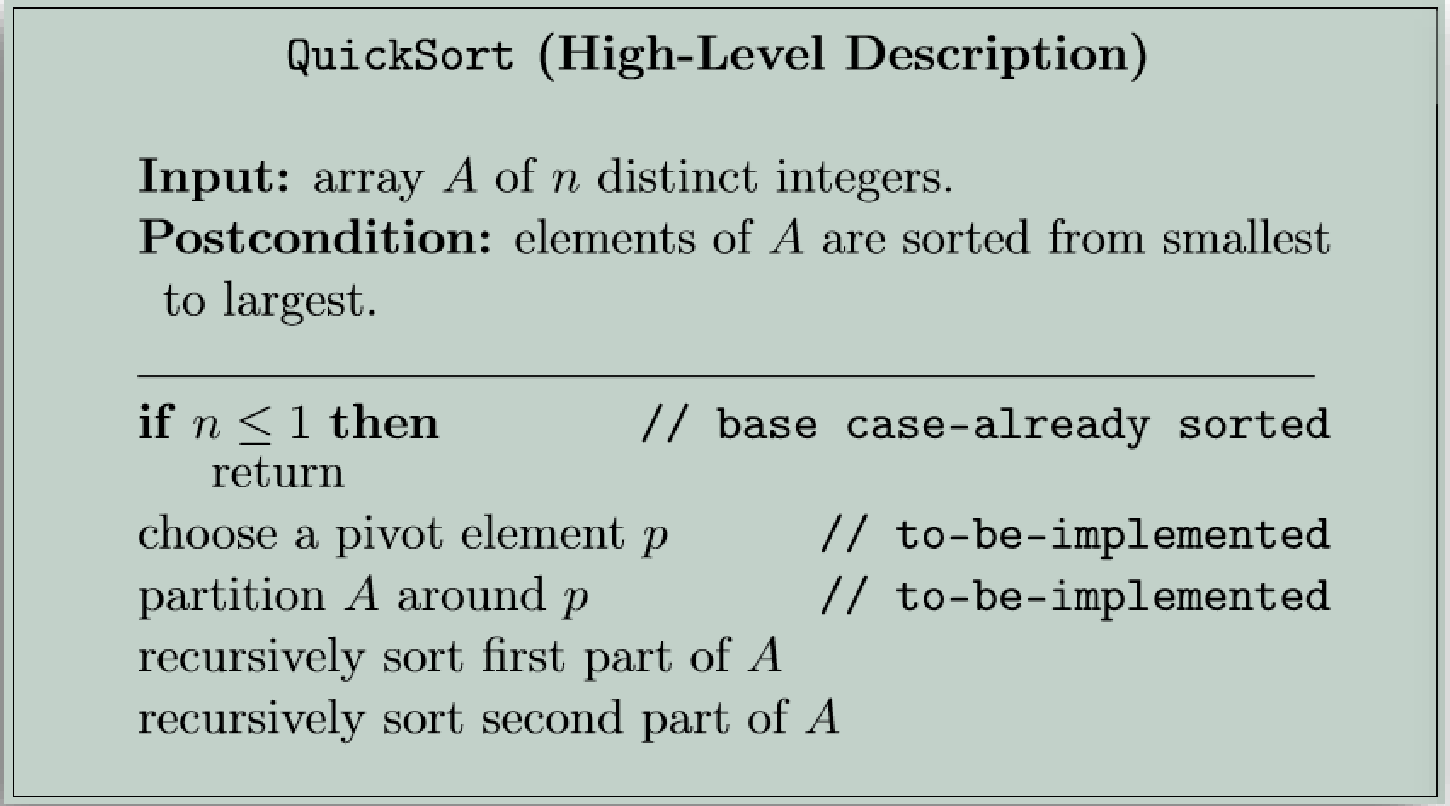


Tony Hoare, 1959

ورودی: یک دنباله از اعداد متمایز با یک ترتیب دلخواه

هدف: مرتب‌سازی دنباله از کوچک به بزرگ

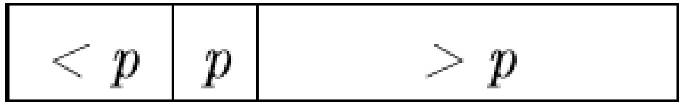
جستجوی سریع (توصیف سطح بالا)



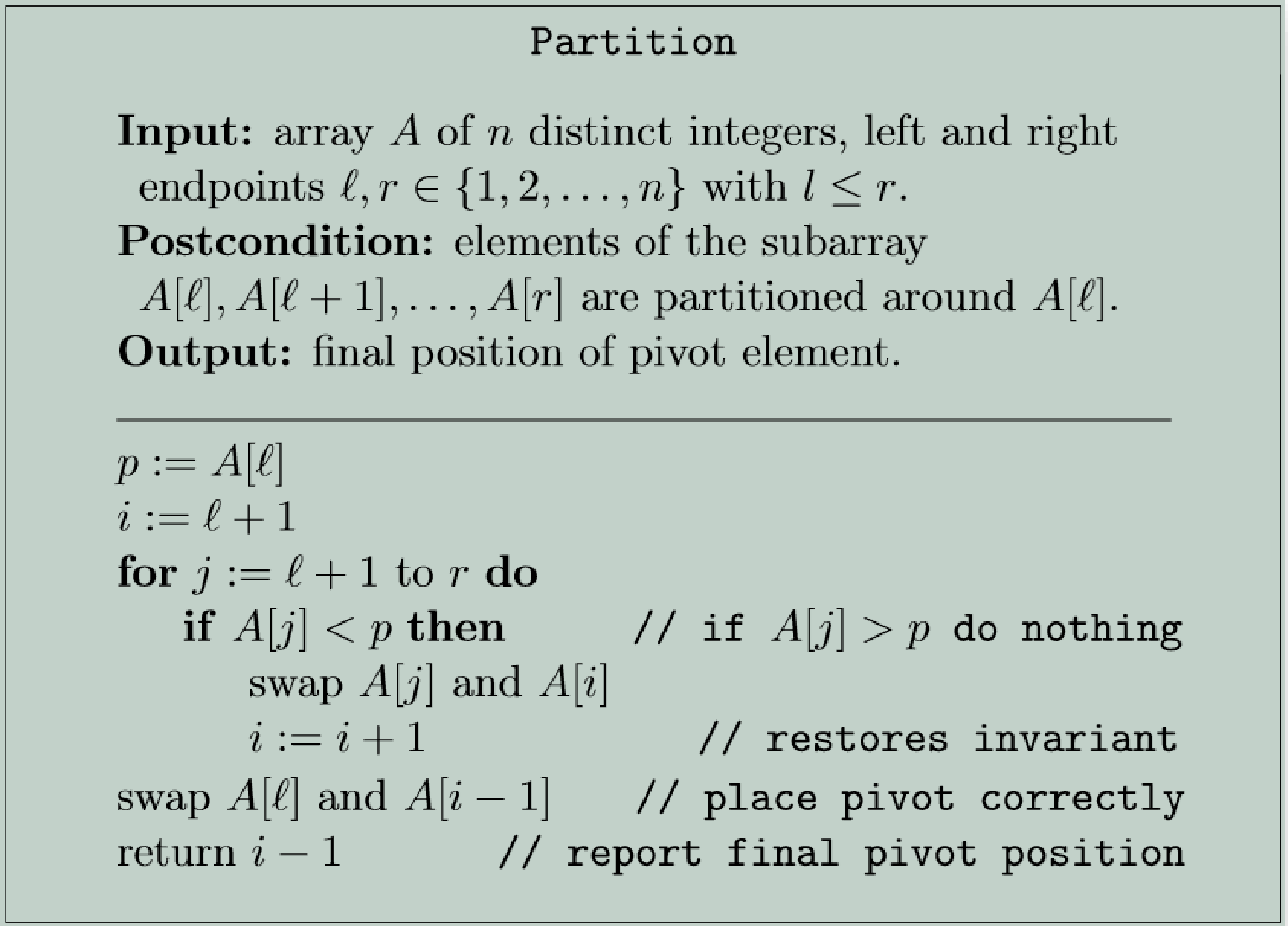
QuickSort (High-Level Description)

Input: array A of n distinct integers.
Postcondition: elements of A are sorted from smallest to largest.

```
if $n \leq 1$ then                   // base case-already sorted
    return
choose a pivot element $p$           // to-be-implemented
partition $A$ around $p$             // to-be-implemented
recursively sort first part of $A$
recursively sort second part of $A$
```



تقسیم به زیرمساله‌های کوچکتر برای جستجوی سریع



Partition

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \dots, n\}$ with $l \leq r$.
Postcondition: elements of the subarray $A[\ell], A[\ell+1], \dots, A[r]$ are partitioned around $A[\ell]$.
Output: final position of pivot element.

```
$p := A[\ell]$
$i := \ell + 1$
for $j := \ell + 1$ to $r$ do
    if $A[j] < p$ then          // if $A[j] > p$ do nothing
        swap $A[j]$ and $A[i]$
        $i := i + 1$                // restores invariant
swap $A[\ell]$ and $A[i-1]$     // place pivot correctly
return $i - 1$          // report final pivot position
```

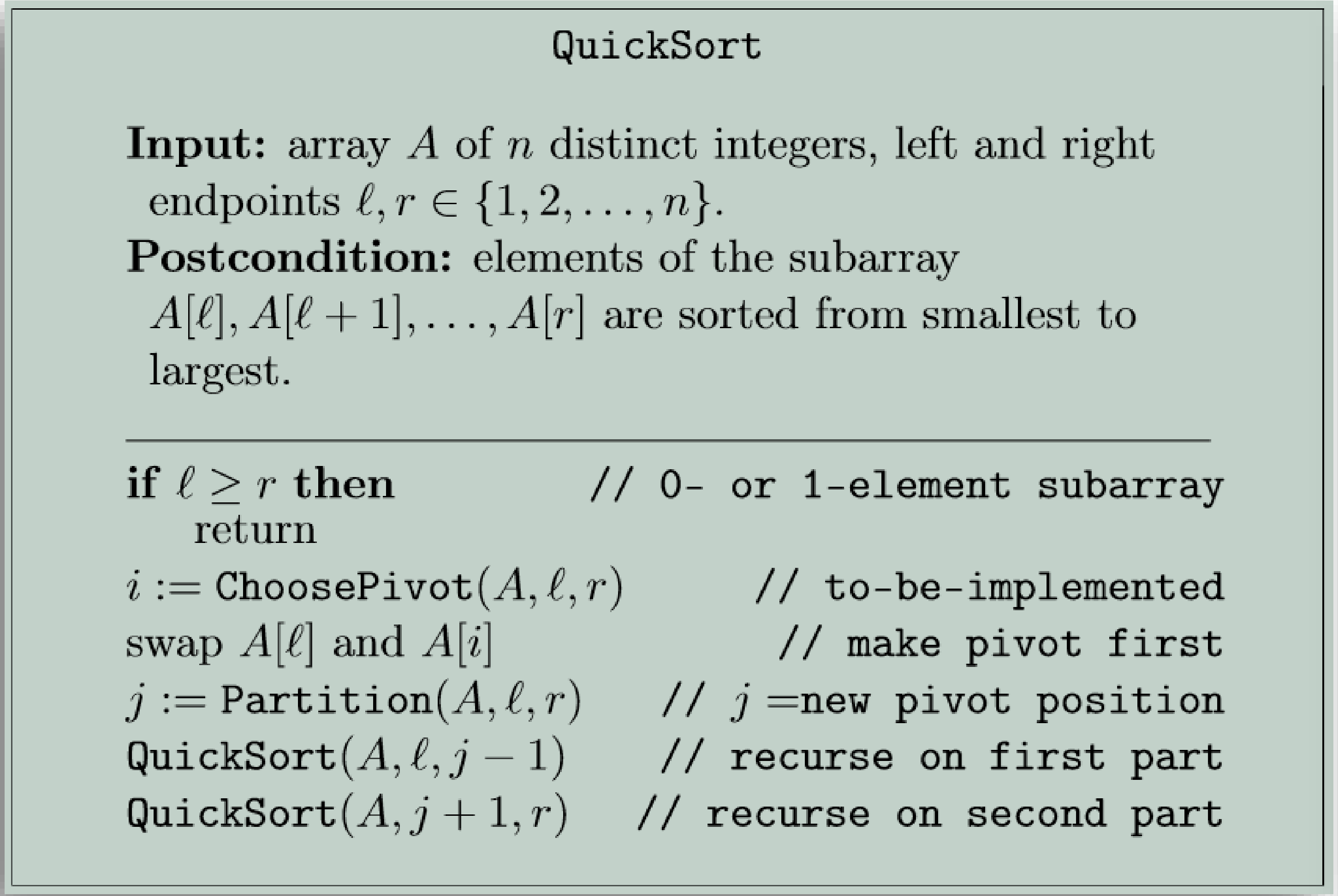
۳	۸	۲	۵	۱	۴	۷	۶

۳	۲	۸	۵	۱	۴	۷	۶

۳	۲	۱	۵	۸	۴	۷	۶

۱	۲	۳	۵	۸	۴	۷	۶

جستجوی سریع



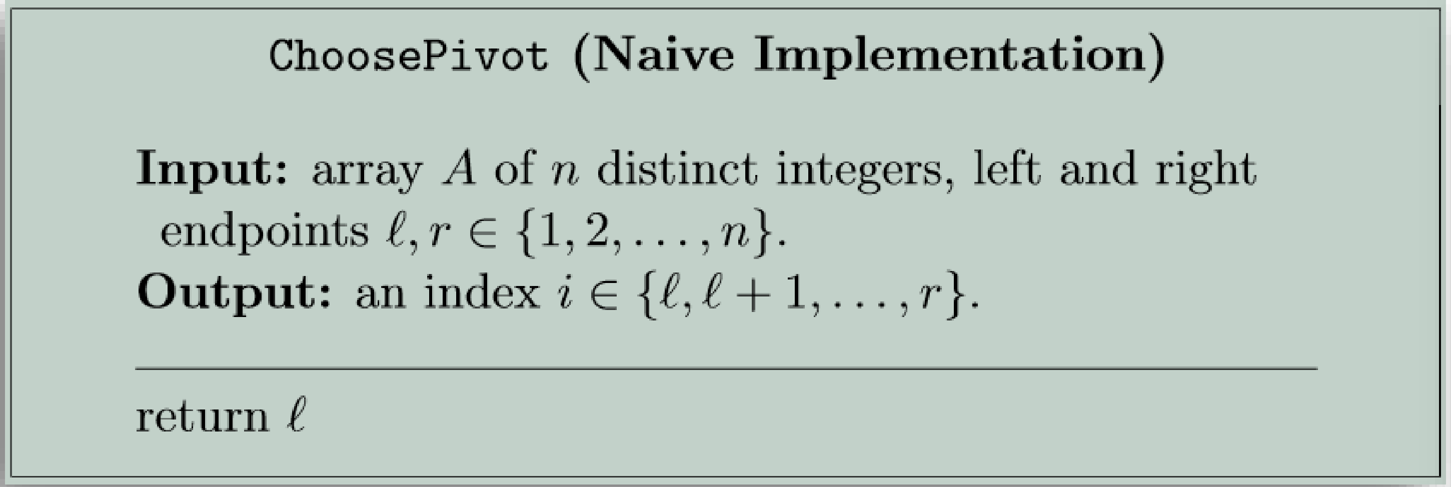
QuickSort

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.

Postcondition: elements of the subarray $A[\ell], A[\ell+1], \ldots, A[r]$ are sorted from smallest to largest.

```
if $\ell \geq r$ then                 // 0- or 1-element subarray
    return
$i :=$ ChoosePivot($A, \ell, r$)      // to-be-implemented
swap $A[\ell]$ and $A[i]$             // make pivot first
$j :=$ Partition($A, \ell, r$)        // $j$ =new pivot position
QuickSort($A, \ell, j-1$)             // recurse on first part
QuickSort($A, j+1, r$)                // recurse on second part
```

انتخاب عنصر راهنما (pivot)

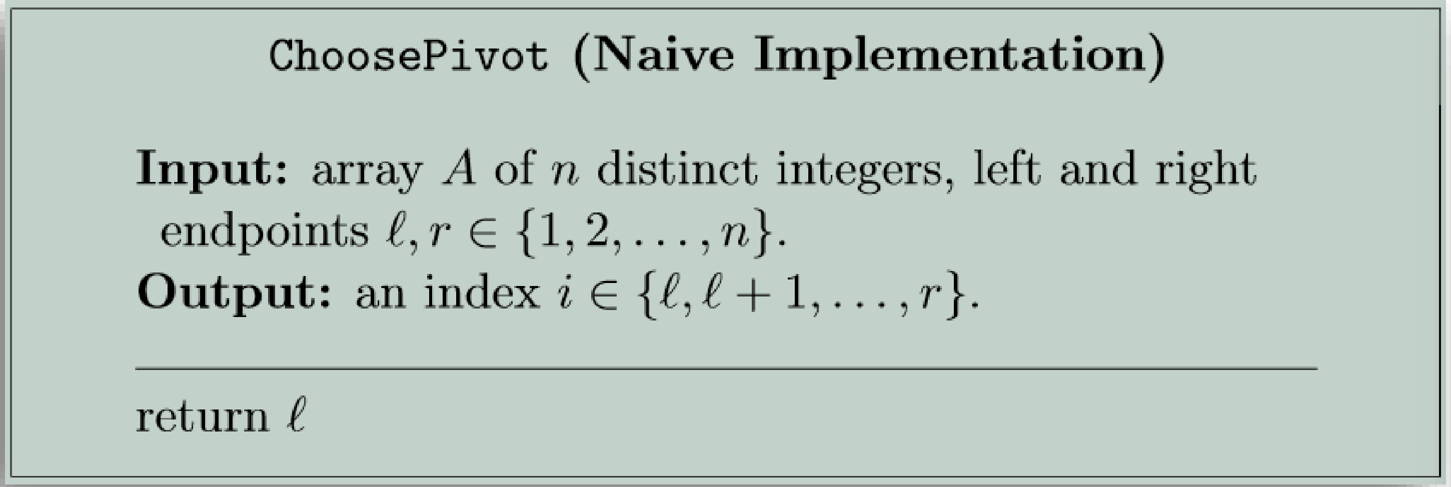


`ChoosePivot` (Naive Implementation)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell + 1, \ldots, r\}$.

```
return $\ell$
```

انتخاب عنصر راهنما (pivot)

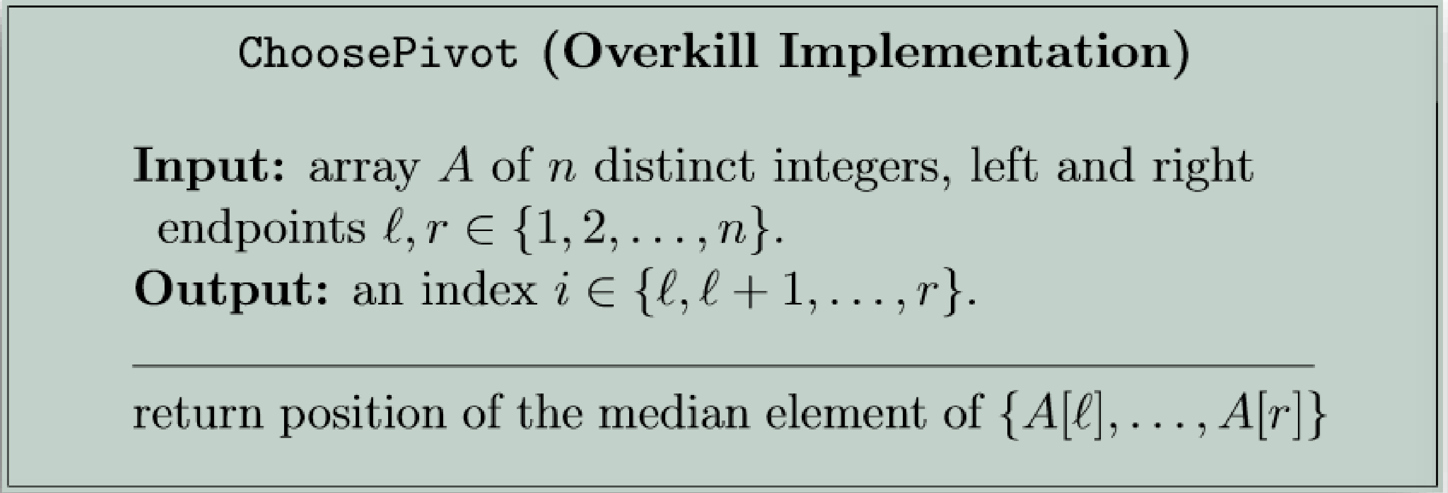


ChoosePivot (Naive Implementation)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell + 1, \ldots, r\}$.

```
return $\ell$
```

انتخاب عنصر راهنما (pivot)



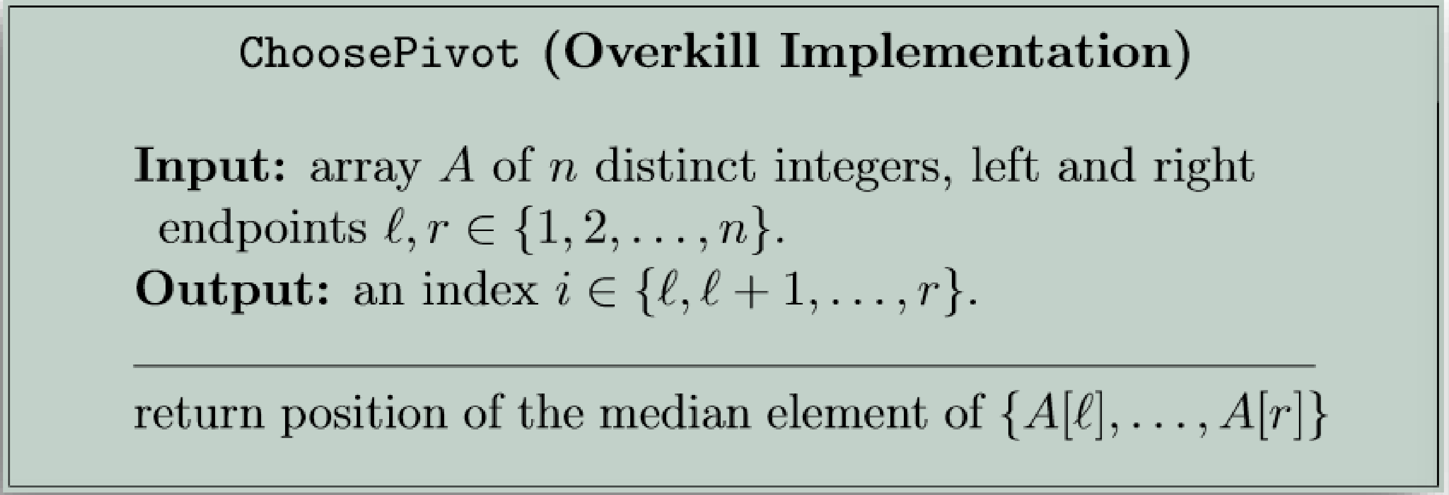
ChoosePivot (Overkill Implementation)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.

Output: an index $i \in \{\ell, \ell + 1, \ldots, r\}$.

```
return position of the median element of $\{A[\ell], \ldots, A[r]\}$
```

انتخاب عنصر راهنما (pivot)

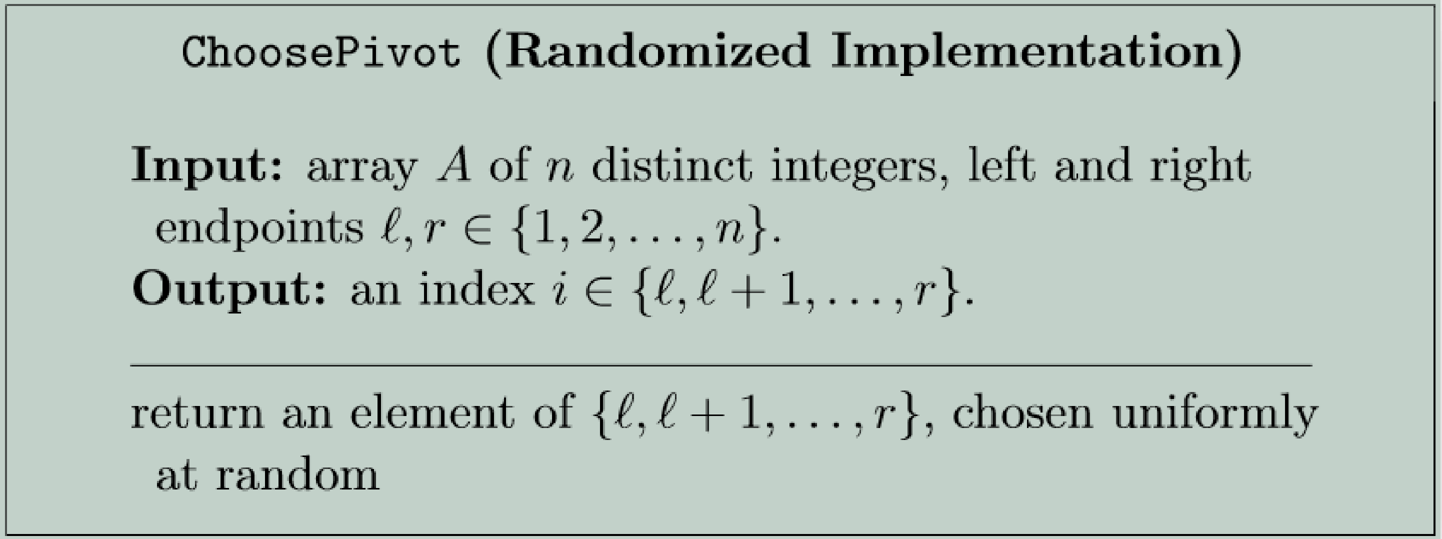


ChoosePivot (**Overkill Implementation**)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell+1, \ldots, r\}$.

return position of the median element of $\{A[\ell], \ldots, A[r]\}$

تحلیل زمانی میانگین

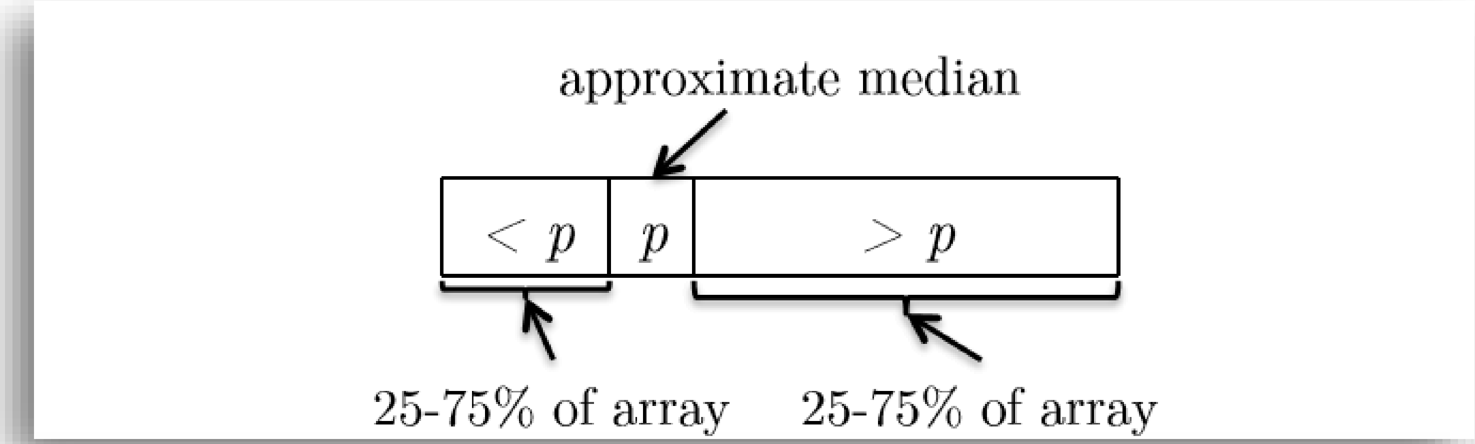


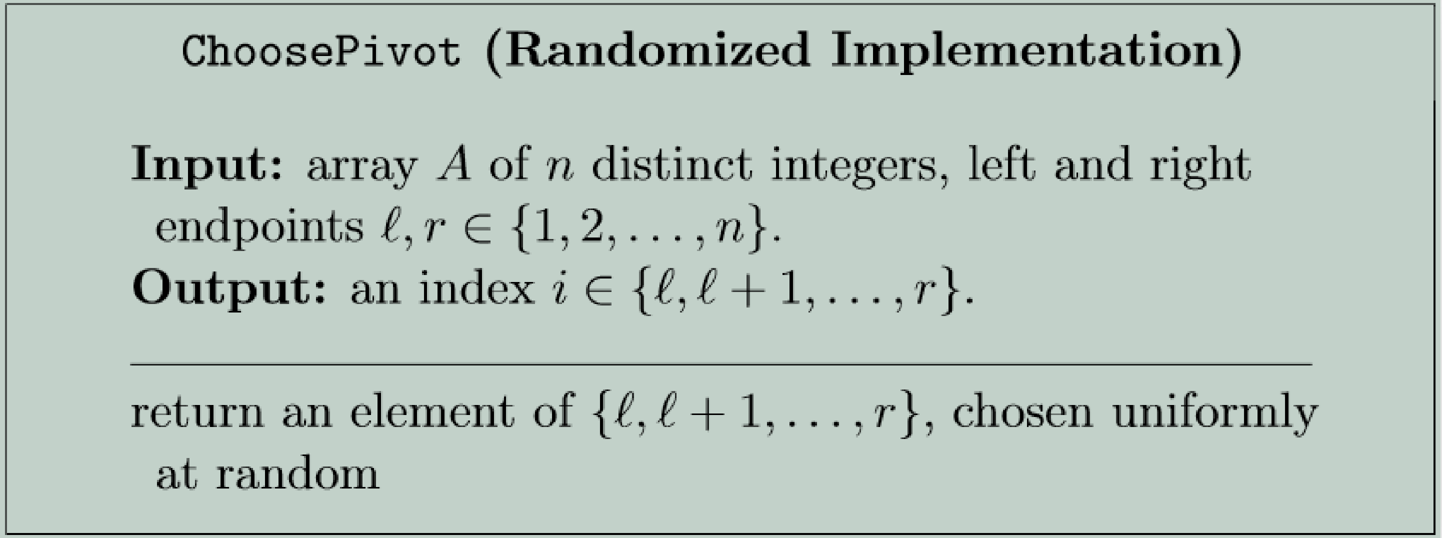
ChoosePivot (Randomized Implementation)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell+1, \ldots, r\}$.

```
return an element of $\{\ell, \ell+1, \ldots, r\}$, chosen uniformly
    at random
```

تحلیل زمانی میانگین





ChoosePivot (Randomized Implementation)

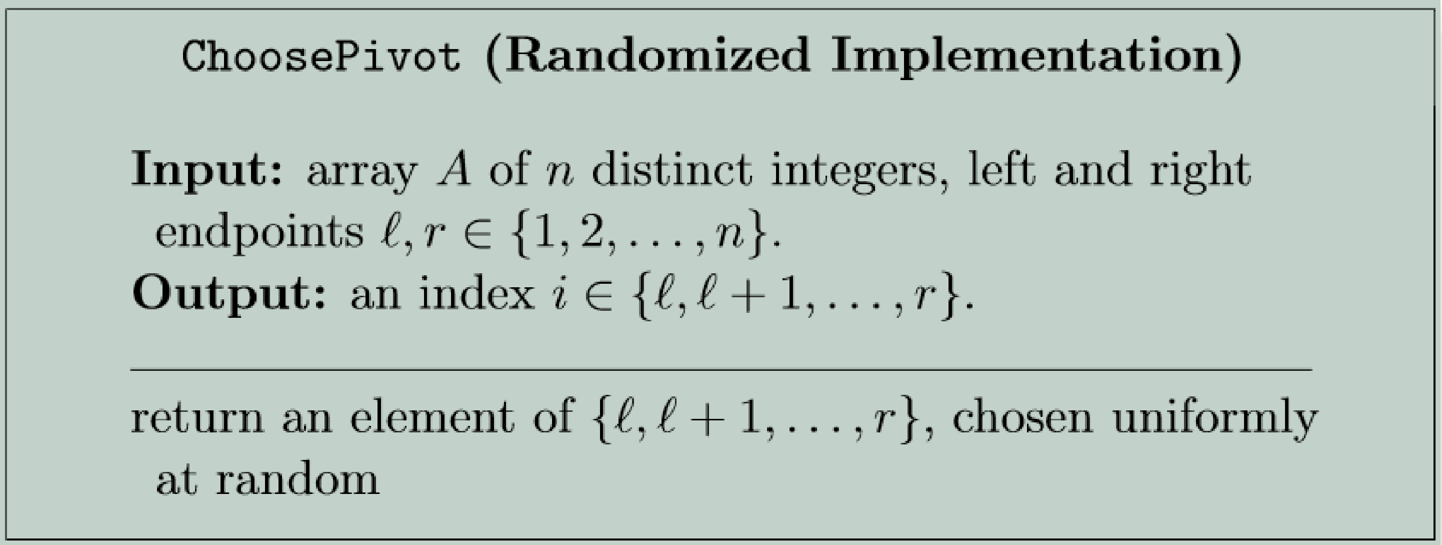
Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell + 1, \ldots, r\}$.

```
return an element of $\{\ell, \ell + 1, \ldots, r\}$, chosen uniformly
  at random
```

تحلیل زمانی میانگین

تحلیل زمانی میانگین

تحلیل زمانی میانگین



ChoosePivot (Randomized Implementation)

Input: array A of n distinct integers, left and right endpoints $\ell, r \in \{1, 2, \ldots, n\}$.
Output: an index $i \in \{\ell, \ell + 1, \ldots, r\}$.

return an element of $\{\ell, \ell + 1, \ldots, r\}$, chosen uniformly at random