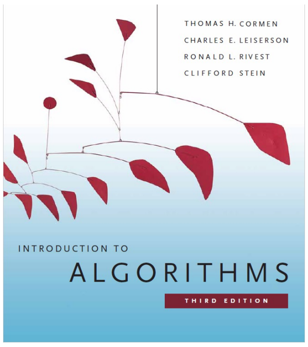
Randomization and Quicksort

Prepared by Shmuel Wimer

Courtesy of Prof. Dror Rawitz



1

Worst Case Run Time

Given **deterministic** algorithm A, let $t_A(x)$ be its run time on input x.

The performance of A w.r.t the input size n, denoted $T_A(n)$ is

$$T_A(n) = \max_{\{x:|x|=n\}} t_A(x)$$

Let $f: \mathbb{N} \to \mathbb{N}$. The run time complexity of A is $O\big(f(n)\big)$ if $\exists n_0 \in \mathbb{N}$ and $\exists C > 0$ s.t. $\forall n \geq n_0$ there is

$$T_A(n) \leq C \cdot f(n)$$

2

Let D_n be the **distribution** of inputs of size n. It is of interest to bound the run time expectation.

$$T_A(n) = \mathrm{E}_{x \sim D_n}[t_A(x)]$$

D_n knowledge is required. Uniform distribution is usually assumed.

Given **probabilistic** algorithm A, let $t_A(x)$ be its run time on input x.

$t_A(x)$ is random variable, depending on A's probabilistic decisions. Hence we consider $\mathrm{E}[t_A(x)]$, and worst case run time of A is

$$T_A(n) = \max_{\{x:|x|=n\}} \mathrm{E}[t_A(x)]$$

Quicksort (C.A.R. Hoare 1962)

Divide-and-conquer algorithm comprising three steps. Given $A[p..r]$,

- **Divide**: Partition $A[p..r]$ into $A[p..q-1]$, $A[q]$, $A[q+1..r]$ s.t. $A[p..q-1] \le A[q] < A[q+1..r]$.
- **Conquer**: Sort $A[p..q-1]$ and $A[q+1..r]$ recursively.
- **Combine**: No work required since $A[p..r]$ is already sorted.

```
QUICKSORT(A, p, r)
   if p < r
       q = PARTITION(A, p, r)
       QUICKSORT(A, p, q − 1)
       QUICKSORT(A, q + 1, r)
```

PARTITION procedure rearranging $A[p..r]$ is crucial, specifically the **pivot** $q \in [p..r]$ choice.

Leftward shift of elements $\leq$ **pivot** $x = A[r]$ and proper reposition of **pivot** next to them ensures that $>$ elements are **rightward**.

$A[p..r]$ before PARTITION	2	8	7	1	3	5	6	4
$A[p..r]$ before pivot reposition	2	1	3	8	7	5	6	4
$A[p..r]$ after pivot reposition	2	1	3	4	7	5	6	8

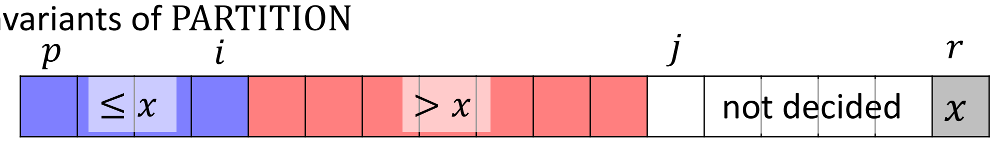
5

PARTITION(A, p, r)

$x = A[r]$ // use last element for pivot
$i = p - 1$ // initialize border of not larger elements
for $j = p$ **to** $r - 1$ // build $A[p..q-1]$
  if $A[j] \leq x$ // not larger than pivot ?
    // yes, move element leftward
    $i = i + 1$ // shift border rightward
    exchange $A[i]$ with $A[j]$
exchange $A[i+1]$ with $A[r]$ // reposition pivot element
return $i + 1$ // return pivot location

6

Invariants of PARTITION



Invariant correction proof.

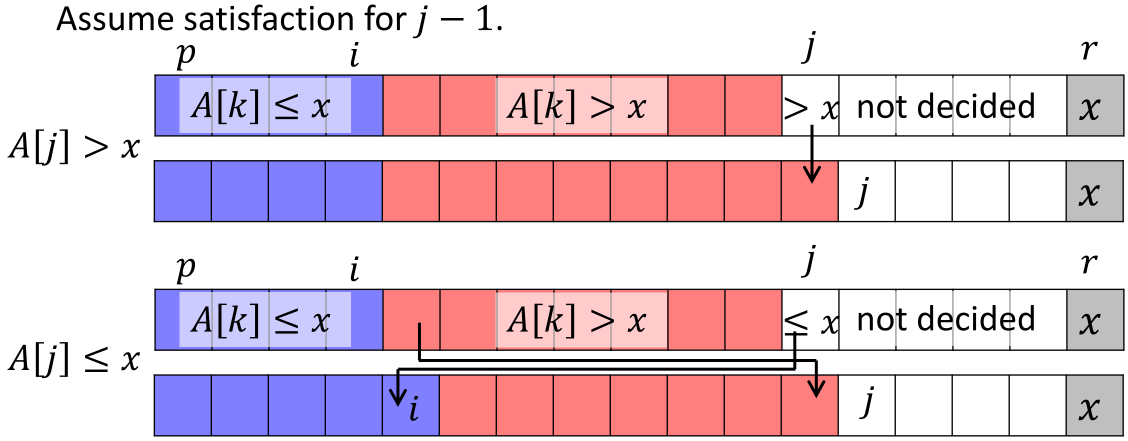
At the beginning of the loop, $\forall k$ there is

1. If $p \leq k \leq i$, then $A[k] \leq x$.
2. If $i + 1 \leq k \leq j - 1$, then $A[k] > x$.
3. If $k = r$, then $A[k] = x$.

The invariant holds prior to each iteration. Proof by induction on j.

At first iteration $i = p - 1$ and $j = p$, $\Rightarrow$ blue and red ranges are empty, $\Rightarrow$ 1 and 2 are trivially satisfied. First line of code satisfies 3.

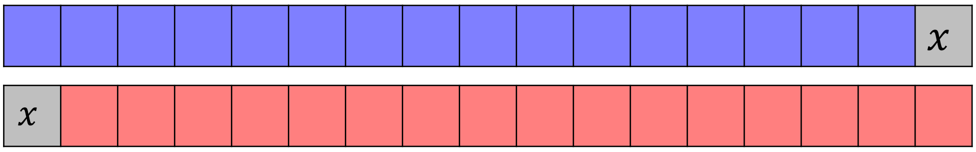
Assume satisfaction for $j - 1$.



Upon termination $j = r \Rightarrow 3$ is satisfied.

PARTITION runtime on $A[p..r]$ is $\Theta(r - p + 1)$. (HW)

Worst-case PARTITION



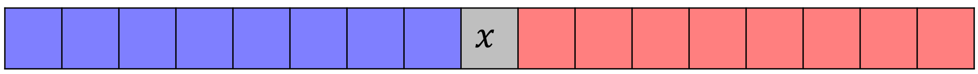
QUICKSORT calls subproblems of size $n - 1$ and 0.

$T(n) = T(n-1) + T(0) + \Theta(n) = T(n-1) + \Theta(n) = \Theta(n^2)$.

(HW: prove $T(n)$, use substitution. What input yields worst-case?)

9

Best-case PARTITION



Results in $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil - 1$ subproblems. Ignoring $\lfloor\ \rfloor$ and $\lceil\ \rceil$, $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$. (HW: prove $T(n)$.)

For any fixed $0 < \alpha \leq 1/2$, if PARTITION always splits into βn and $(1-\beta)n - 1$, $\alpha \leq \beta \leq 1/2$, $T(n) = \Theta(n \log n)$. (HW: prove.)

To overcome the worst-case input, pivot can be chosen randomly, expecting average partition to be reasonably balanced.

10

RANDOMIZED-PARTITION(A, p, r)

i = RANDOM(p, r) // pivot drawn randomly
exchange $A[r]$ with $A[i]$ // make pivot rightmost
return PARTITION(A, p, r) // deterministic

RANDOMIZED-QUICKSORT(A, p, r)

if $p < r$
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT$(A, p, q - 1)$
    RANDOMIZED-QUICKSORT$(A, q + 1, r)$

Run time analysis – method 1

Assume w.l.o.g that all A's elements are different.

Let $\Theta(n) = an$ be the runtime of RANDOMIZED−PARTITION, which may choose any of the n element with equal probability. $\Rightarrow$

(1) $T(n) = an + \frac{1}{n}\sum_{i=0}^{n-1}\big(T(i) + T(n-i-1)\big) = an + \frac{2}{n}\sum_{i=0}^{n-1} T(i)$.

Multiplying (1) by $n \Rightarrow$

(2) $nT(n) = an^2 + 2\sum_{i=0}^{n-1} T(i)$.

Substitution $n-1$ into (2) $\Rightarrow$

(3) $(n-1)T(n-1) = a(n-1)^2 + 2\sum_{i=0}^{n-2} T(i)$.

Subtraction of (3) from (2) $\Rightarrow$

$nT(n) = (n+1)T(n-1) + a(2n-1)$

$\leq (n+1)T(n-1) + 2a(n+1)$.

Division by $n(n+1) \Rightarrow$

(4) $T(n)/(n+1) \leq T(n-1)/n + 2a/n$.

Substitution $F(n) = T(n)/(n+1)$ in (4) $\Rightarrow$

(5) $F(n) \leq 2a/n + F(n-1)$

$\leq 2a/n + 2a/(n-1) + F(n-2)$

$\leq 2a/n + 2a/(n-1) + 2a/(n-2) + F(n-3) \leq \cdots$

R.H.S of (5) summed to

$F(n) \leq 2a\sum_{i=1}^{n}\frac{1}{i} = 2aH_n$, where H_n is the nth harmonic sum. $\Rightarrow$

$T(n) = 2a(n+1)H_n = O(n \log n)$. ■

Run time analysis – method 2

$\mathrm{RANDOMIZED-PARTITION}$ dominates runtime, called at most n times since element can be pivot at most once.

Within $\mathrm{PARTITION}$ loop there are comparisons $A[j] \leq x$ of $A[j]$ with the pivot x.

Let X be total number comparisons **over all** $\mathrm{PARTITION}$'s loops. Total $\mathrm{QUICKSORT}$ runtime is therefore $O(n + X)$.

So what is X?

Rename the elements of A as $z_1, z_2, \dots, z_n$ by their ascending sorted values and let $Z_{ij} = \{z_i,\ z_{i+1}, \dots, z_j\}$.

How many times $\mathrm{QUICKSORT}$ compares z_i and z_j, $i \neq j$?

In $\mathrm{PARTITION}$'s comparison $z \leq x$, x is the pivot, and once the loop completes x is placed in final location and **never touched again**.

Hence $\mathrm{QUICKSORT}$ compares any z_i and z_j, $i \neq j$, at most once.

Define random variable $X_{ij} = \begin{cases} 1 & z_i \text{ is compared to } z_j \\ 0 & \text{otherwise} \end{cases}.$

Then, $X = \sum_{1 \le i < j \le n} X_{ij} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$, and

expectation linearity

(6) $\mathrm{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{E}[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[z_i\, , z_j \text{ compared}].$

$\mathrm{RANDOMIZED-PARTITION}$ chooses pivot x from subarray $A[p..r]$ randomly and independently.

Let $x \in Z_{ij}$. There are 3 cases:

- $z_i < x < z_j$. z_i and z_j never compared since x splits Z_{ij} into smaller and larger parts.

z_i					x				z_j

- $x = z_i$. x is compared to all, hence z_i and z_j are compared.

z_i									z_j

- $x = z_j$. x is compared to all, hence z_i and z_j are compared.

z_i									z_j

length of $[z_i, \ldots, z_j]$

x chosen at random $\Rightarrow \Pr[z_i \text{ is compared to } z_j] = 2/(j - i + 1)$.

only $x = z_i$ and $x = z_j$ apply

Substitution into (6) yields

$k = j - i$

$\mathrm{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{j-i=1}^{n-i} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} <$

$\sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} = 2 \sum_{i=1}^{n-1} H_n = O(n \log n)$. ■

harmonic sum