ITMO UNIVERSITY

How to Win Coding Competitions: Secrets of Champions

Week 3: Sorting and Search Algorithms
Lecture 5: Quicksort Modifications

Maxim Buzdalov
Saint Petersburg 2016

Quicksort

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[(s+e)/2]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Recall quicksort. . .

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[(s+e)/2]$
    while $s' \leq e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \leq e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \leq e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \leq e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Why using $(s+e)/2$?

```
procedure QUICKSORT($A, \prec, s, e$)
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[(s+e)/2]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then QUICKSORT($A, \prec, s, e'$) end if
    if $s' \le e$ then QUICKSORT($A, \prec, s', e$) end if
end procedure
```

Why using $(s+e)/2$?

- Some sources suggest either s or e

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[(s+e)/2]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \le e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

Why using $(s+e)/2$?

- Some sources suggest either s or e
- But this choice leads to $\Theta(N^2)$ on sorted arrays!

Quicksort

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[(s+e)/2]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Why using $(s+e)/2$?

- Some sources suggest either s or e
- But this choice leads to $\Theta(N^2)$ on sorted arrays!
- $(s+e)/2$ makes it fast on such arrays

Quicksort

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[(s+e)/2]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Why using $(s+e)/2$?

- Some sources suggest either s or e
- But this choice leads to $\Theta(N^2)$ on sorted arrays!
- $(s+e)/2$ makes it fast on such arrays
- But is it the only choice?

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Modification: Random pivot

Quicksort

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \le e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

Modification: Random pivot

- Turns average $O(N \log N)$ time on random input into expected $O(N \log N)$ time on any input

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

Modification: Random pivot

- Turns average $O(N \log N)$ time on random input into expected $O(N \log N)$ time on any input
- Only $O(N)$ queries to random number generator

Quicksort

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \le e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

Modification: Random pivot

- Turns average $O(N \log N)$ time on random input into expected $O(N \log N)$ time on any input
- Only $O(N)$ queries to random number generator
- Efficient in practice

```
procedure $\textsc{Quicksort}(A, \prec, s, e)$
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then $\textsc{Quicksort}(A, \prec, s, e')$ end if
    if $s' \le e$ then $\textsc{Quicksort}(A, \prec, s', e)$ end if
end procedure
```

K-th order statistic:

- Find what will be the k-th element of the array, if it was sorted

Finding k-th order statistic

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \leq e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \leq e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \leq e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \leq e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

K-th order statistic:

- Find what will be the k-th element of the array, if it was sorted
- Faster than sorting!

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \leq e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \leq e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \leq e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \leq e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

Solution: Modify quicksort!

Finding k-th order statistic

```
procedure QUICKSORT($A$, $\prec$, $s$, $e$)
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ then QUICKSORT($A$, $\prec$, $s$, $e'$) end if
    if $s' \le e$ then QUICKSORT($A$, $\prec$, $s'$, $e$) end if
end procedure
```

Solution: Modify quicksort!

- Run only half of it!
  - Don't sort the part which is not needed

```
procedure $\textsc{Quicksort}(A, \prec, s, e, k)$
    $s' \leftarrow s$, $e' \leftarrow e$, $M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1$, $e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ and $k \le e'$ then $\textsc{Quicksort}(A, \prec, s, e', k)$ end if
    if $s' \le e$ and $s' \le k$ then $\textsc{Quicksort}(A, \prec, s', e, k)$ end if
end procedure
```

Solution: Modify quicksort!

- Run only half of it!
  - Don't sort the part which is not needed

Finding k-th order statistic

```
procedure $\textsc{Quicksort}(A, \prec, s, e, k)$
    $s' \leftarrow s,\ e' \leftarrow e,\ M \leftarrow A[\textsc{Random}(s, e)]$
    while $s' \le e'$ do
        while $A[s'] \prec M$ do $s' \leftarrow s' + 1$ end while
        while $M \prec A[e']$ do $e' \leftarrow e' - 1$ end while
        if $s' \le e'$ then
            $A[s'] \Leftrightarrow A[e']$
            $s' \leftarrow s' + 1,\ e' \leftarrow e' - 1$
        end if
    end while
    if $s \le e'$ and $k \le e'$ then $\textsc{Quicksort}(A, \prec, s, e', k)$ end if
    if $s' \le e$ and $s' \le k$ then $\textsc{Quicksort}(A, \prec, s', e, k)$ end if
end procedure
```

Solution: Modify quicksort!

- Run only half of it!
  - Don't sort the part which is not needed
- Running time: $\Theta(N)$ in expectation
  - Array size reduction as in quicksort