CS3000

5/16 - Tue.

Admin

- Long HW1 due 9pm
- Short HW2 out, due 5/18 9pm
- Today: rec 2!

Agenda

1. Quicksort overview
2. Quicksort code
3. Quicksort correctness

$\boxed{\text{Recap}}$

- Func (A, p, r)
  - what does p rep? → left index
  - what does r rep? → right index

    A[p..r]
  - compute midpoint? → $\left\lfloor \frac{p+r}{2} \right\rfloor$
  - what do we know if $p<r$? → subarray has length >1

1. Quicksort

sorting: selection sort, $\Theta(n^2)$

mergesort $T(n) = 2 \cdot T(n/2) + n$

$\Theta(n \lg n)$

quicksort

D & C: Binary search, Karatsuba, mergesort

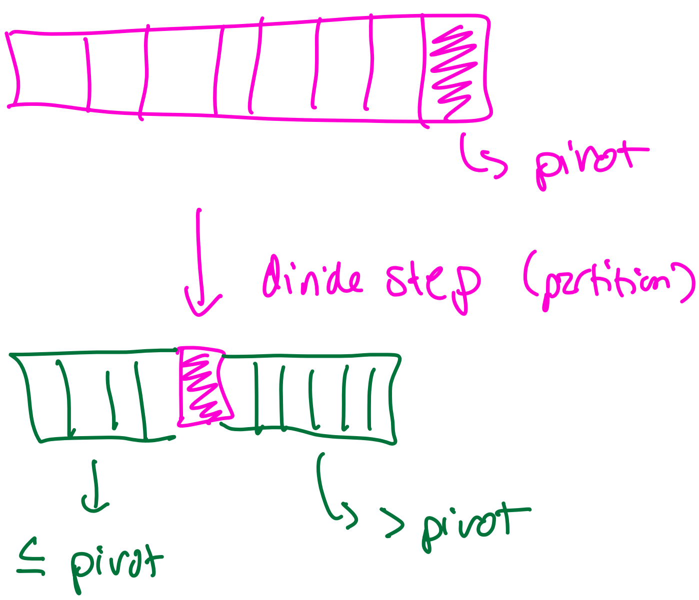
quicksort

Divide: choose a pivot (an element from A)
put everything $\leq$ pivot on its left, $>$ pivot on right

Conquer: recursively sort left, right halves
(not the pivot!)

Combine: (no op)

Array:



After this step maybe we have

8	5	12	18	22	21	30

↳ pivot!

(never has to move)

next, recursively sort

8	5	12	18	22	21	30

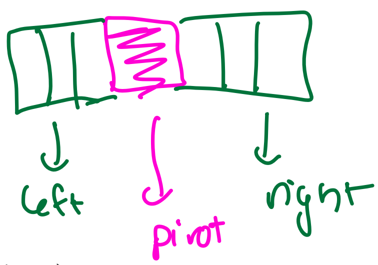
↳ sort left

↳ sort right

↳ ignore pivot!

$\boxed{\text{Best case:}}$ balance around the pivot

(pivot is median)



$\boxed{\text{Worst case:}}$ unbalanced around pivot

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

left (empty)

pivot

↳ right

luck!

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        swap A[i], A[j]
swap A[i + 1], A[r]
return i + 1
```

2. Quicksort Pseudocode

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

→ partition around pivot

→ recursively sort left half, right half

$\boxed{A[q] \text{ is done!}}$

```
PARTITION(A, p, r)
  x = A[r]
  i = p - 1
  for j = p to r - 1
      if A[j] ≤ x
          i = i + 1
          swap A[i], A[j]
  swap A[i + 1], A[r]
  return i + 1
```

→ choose pivot

put everything ≤ pivot on left

put everything > pivot on right

returns location of pivot

- walk through partition for one call of array A
- $i = p - 1$ } two indices into the array
- $j = p$
- where are the elements > pivot?

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

```
PARTITION(A, p, r)
  x = A[r]
  i = p - 1
  for j = p to r - 1
      if A[j] ≤ x
          i = i + 1
          swap A[i], A[j]
  swap A[i + 1], A[r]
  return i + 1
```

$A = \langle 2, 9, 10, 3, 12, 18, 5 \rangle$

↳ pivot

$p = 1$

$r = 7$

$\boxed{x = 5}$

$i = p - 1 = 0$

① $j = p = 1$

$A[j] \le x$?

$i = 1$

swap $A[1], A[1]$

$\langle 2, 9, 10, 3, 12, 18, 5 \rangle$

$\boxed{\begin{matrix} i = 1 \\ j = 1 \end{matrix}}$

(2) $j = 2$

$A[j] \le x$?

i stays the same

at end loop, $j \mathrel{+}= 1$

(3) $j = 3$

$\langle 2, 9, 10, 3, 12, 18, 5 \rangle$

$i=1$ $j=3$

$A[j] \le x$?

i stays the same

at end of loop, $j \mathrel{+}= 1$

$>$ pivot

$\langle 2, 9, 10, 3, 12, 18, 5 \rangle$

$i=1$ $j=4$

(4) $j = 4$

$A[j] \le x$?

$i = 2$

swap $A[i], A[j]$

at end of loop, $j \mathrel{+}= 1$

$\langle 2, 3, 10, 9, 12, 18, 5 \rangle$

$i=2$ $j=5$

(5) $j = 5$

$A[j] \le x$? no

i stays the same

at end of loop, $j \mathrel{+}= 1$

$\langle 2, 3, 10, 9, 12, 18, 5 \rangle$

$i=2$ $j=6$

(6) $j = 6$

$A[j] \le x$? no

i stays the same

at end of loop $j \mathrel{+}= 1$

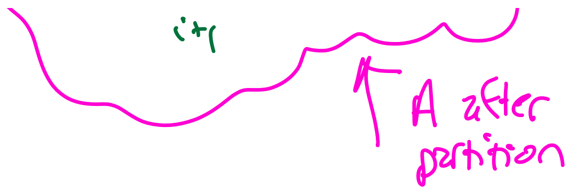
$\langle 2, 3, 10, 9, 12, 18, 5 \rangle$

$i=2$ $j=7$

swap $A[i+1], A[r]$

$\langle 2, 3, \underline{5}, 9, 12, 18, 10 \rangle$

return i+1 ⟶ $\boxed{\text{position 3}}$



- Is everything left of 5 ≤ 5? yes!
- Is everything right of 5 > 5? yes!
- returns position of pivot (3)
- pivot never moves!

10:51

3. Quicksort Correctness

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

PARTITION(A, p, r)
```
1  x = A[r]
2  i = p − 1
3  for j = p to r − 1
4      if A[j] ≤ x
5          i = i + 1
6          swap A[i], A[j]
7  swap A[i + 1], A[r]
8  return i + 1
```

⟶ partition step

Where are the elements greater than the pivot?

start : i+1
end : j−1
} at start of each iteration (lines 3-6)

Loop invariant for any index k in array A, <u>if</u> $i+1 \le k \le j-1$, <u>then</u> $A[k] > x$

1. Init. $i = p-1$ $i+1$ to $j-1$
$j = p$ p $p-1$

no indices between $p, p-1$. Trivially true

2. Maintenance

- Assume true for up to $j-1$

Case one $A[j] > x$

nothing happens!
except, j increments at very end
Before this iteration $A[i+1 \dots j-1] > x$
At end, we have $A[j] > x$ $A[i+1 \dots j] > x$
Finally, $j = j+1$, so the invariant hold

Case two $A[j] \le x$

$i = i+1$
swap $A[i], A[j]$ → $A[j] \le x$
↳ $A[i]$ must be $> x$
Finally, $j = j+1$, so the invariant holds

3. Termination:

$j = r$ after the loop ends
By loop invariant, $A[i+1 \dots r-1]$ is $> x$
↳ up to but not including the pivot is bigger

$Fun(x, n)$ $\left(x^{n-1}, x^{n}\right)$