

- ① Analysis Avg Case Run-Time Quick Sort
- ② Quickselect (what it is), Analysis
- ③ More Sorting [Shell sort, Counting, Radix, Proof lower bound sorting]

QS



Split prob

<u>L</u>	<u>R</u>	
0	n-1	$\frac{1}{n}$
1	n-2	$\frac{1}{n}$
2	n-3	$\frac{1}{n}$
3	n-4	$\frac{1}{n}$
⋮		
n-1	0	



$$T(n) = T(l) + T(r) + n$$

$$\begin{aligned}
 T(n) &= \frac{1}{n} (T(0) + T(n-1) + n) \\
 &+ \frac{1}{n} (T(1) + T(n-2) + n) \\
 &+ \frac{1}{n} (T(2) + T(n-3) + n) \\
 &+ \dots \\
 &+ \frac{1}{n} (T(n-1) + T(0) + n)
 \end{aligned}$$

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

$$nT(n) = \left[2 \sum_{i=0}^{n-1} T(i) \right] + n^2$$

$$n T(n) = \left[2 \sum_{i=0}^{n-1} T(i) \right] + n^2$$

~~$n^2 - 2nT$~~

$$- \frac{(n-1) T(n-1)}{\cancel{(n-1) T(n)}} = \left[2 \sum_{i=0}^{n-2} T(i) \right] + (n-1)^2$$

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

$$n T(n) = \underline{(n-1) T(n-1)} + \underline{2 T(n-1)} + (2n-1)$$

$$\frac{n T(n)}{n(n+1)} = \frac{(n+1) T(n-1)}{n(n+1)} + \frac{(2n-1)}{n(n+1)}$$

$$\frac{T(n)}{n+1} \approx \frac{T(n-1)}{n} + \frac{2}{n+1}$$

(approx that doesn't change the end ans too much)

$$\text{Let } S(n) = \frac{T(n)}{n+1}$$

$$\boxed{T(1) = 1}$$

$$S(1) = \frac{1}{2}$$

$$\begin{aligned} S(n) &= \underline{S(n-1)} + \frac{2}{n+1} \\ &= S(n-2) + \frac{2}{n} + \frac{2}{n+1} \\ &= S(n-3) + \frac{2}{n-1} + \frac{2}{n} + \frac{2}{n+1} \end{aligned}$$

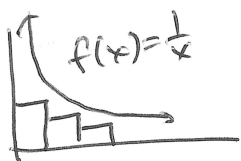
$$= S(1) + \frac{2}{3} + \frac{2}{4} + \dots + \frac{2}{n+1}$$

$$= \frac{1}{2} + \frac{2}{3} + \frac{2}{4} + \dots + \frac{2}{n+1}$$

$$= \left(\sum_{i=2}^{n+1} \frac{2}{i} \right) - \frac{1}{2}$$

$2 \ln(n+1)$ and
 $2 \ln(n) + 2$

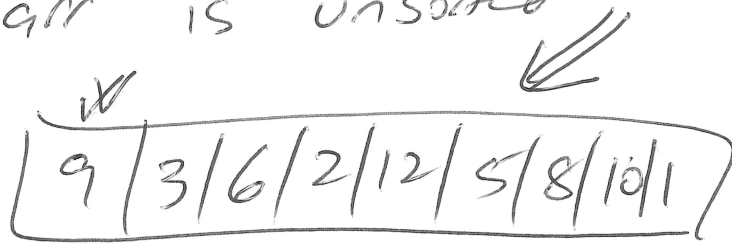
$$S(n) = O(\lg n) \Rightarrow T(n) = O(n \lg n)$$



$$H_n = \sum_{i=1}^n \frac{1}{i} \sim \ln n$$

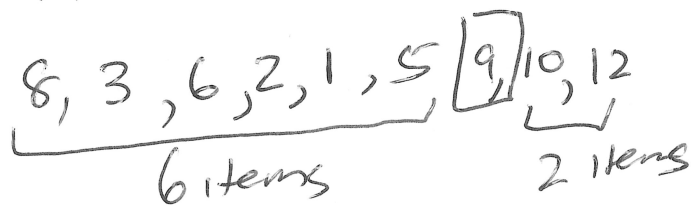
Quickselect (int[] arr, int rank)

Question what is the rank smallest item in arr? (arr is unsorted)



4th

① Partition



② let $x = \#$ items left partition

if rank $\leq x$

return quickselect (on left array)

else if rank == $x+1$

return partition element

else return quickselect (right array, rank - $x - 1$)

Avg Case Run Time

Worst Case = $O(n^2)$

Best Case = $O(n)$ first time

My Homework for 2/15/2024

(1) Type Quick Select Analysis

(2) Fix Sorts.java