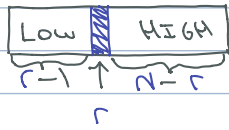


# Quick-sort worst-case

## How to quicksort

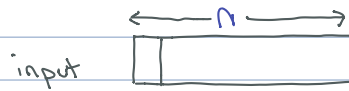
### 1. Row partition



### 2. Run Quicksort on low

### 3. Run Quicksort on high

## Partition



$O(N)$  time

EG. 7 2 4 | 8 | 9 16  
Low HIGH  
 $r = 4$

All  $\in$  Low  $< r$   
All  $\in$  HIGH  $> r$

## Step 1. Write a formula

$$T_w(N) = \max_{1 \leq r \leq N} \left\{ \underset{\substack{\uparrow \\ \text{time for \#2}}}{T_w(r-1)} + \underset{\substack{\uparrow \\ \text{time for \#3}}}{T_w(N-r)} + \underset{\substack{\uparrow \\ \text{time for \#1}}}{O(N)} \right\}$$

## Step 2. Guess $T_w(N) = O(N^2) \leq cN^2$ for some $c > 0$

## Step 3. Check

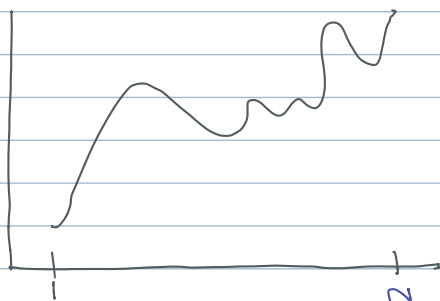
$$T_w(N) = \max_{1 \leq r \leq N} \left\{ T_w(r-1) + T_w(N-r) + O(N) \right\}$$

Induction Hypothesis:  $T_w(i) \leq ci^2$  (from guess)

$$T_w(N) = \max_{1 \leq r \leq N} \left\{ c(r-1)^2 + c(N-r)^2 + a \cdot n \right\}$$

Need to find:

$$\max_{1 \leq r \leq N} F(N)$$



$$= 2C(r-1) - 2C(n-r) = 0$$

$$r \approx \frac{n}{2}$$

$$\max F(r) = \max \{ F(1), F(\frac{n}{2}), F(n) \}$$

$$= \max \{ C(n-1)^2 + a \cdot n, 2C(\frac{n}{2})^2 + a \cdot n \}$$

$$= C(n-1)^2 + a \cdot n$$

$$= C(n^2 - 2n + 1) + a \cdot n$$

...

$$= Cn^2 \text{ when } C \gg a$$

Must be able to do so in 7 minutes on midterm