Lecture 4

- Finish Sorting + Medians

OH

- Tue Night in campus (Virgil + Nitish) 7:30pm
- 2 OH online
- UG/Align Tutorial Anirudh Thu

+ UG/Align OH Fri midday 'Laxmi'

No HW ←

HW 4,5,6 Virgil OH Fri 6pm Teams

QS(A[b:e])

  p = Partition(A[b:e])

  rearranges A $[\le v \mid \overset{p}{v} \mid > v]$

  QS(A[b:p-1])

  QS(A[p+1:e])

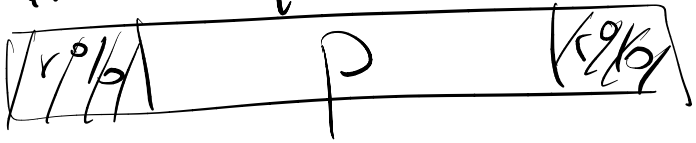
$T(n) = T(p) + T(n-p-1) + n$

best case $T(n) = 2T\left(\frac{n}{2}\right) + n$

$\boxed{\Theta(n \log n)}$

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

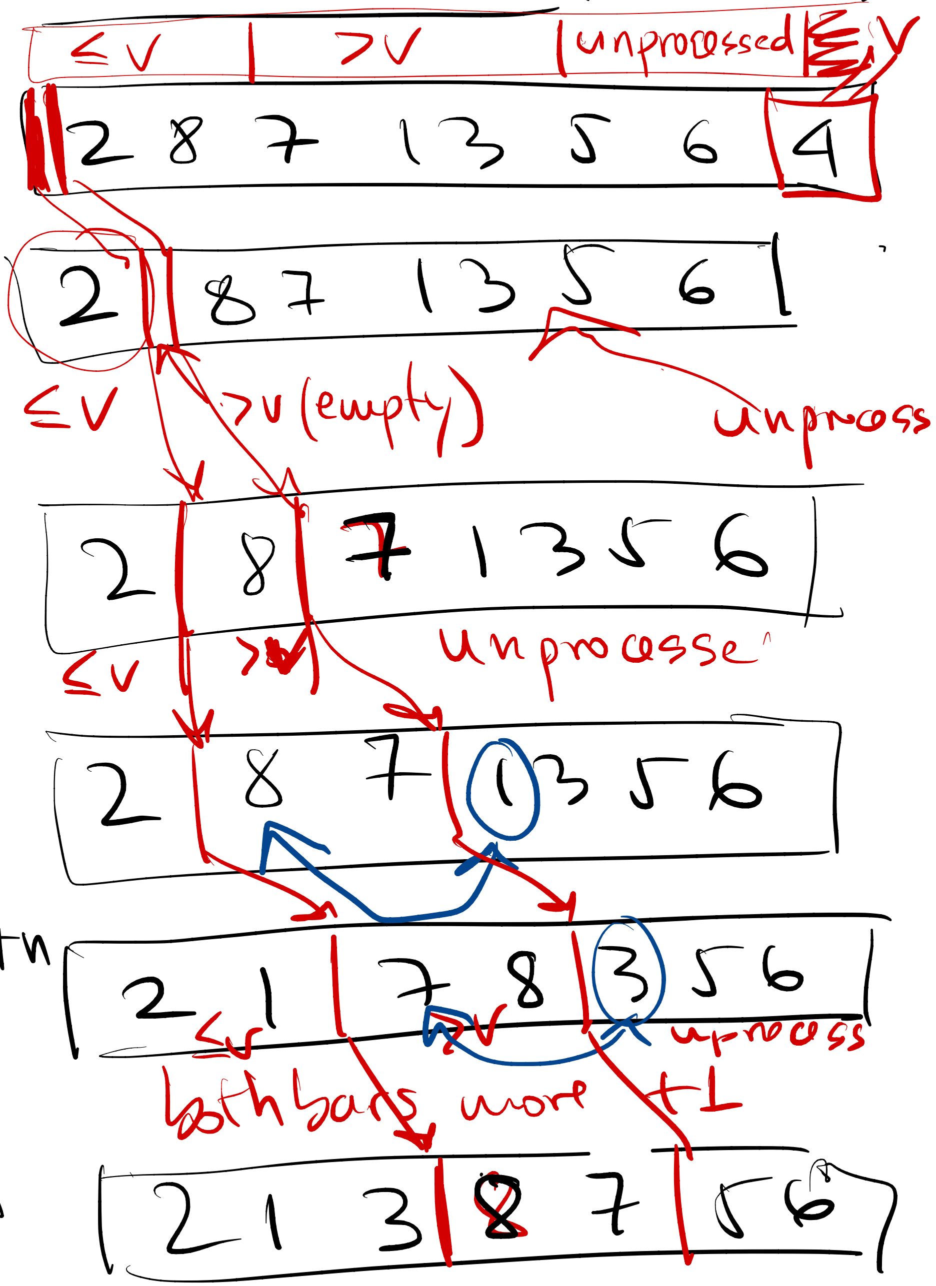
$10\% n \le p \le n - 10\% n$

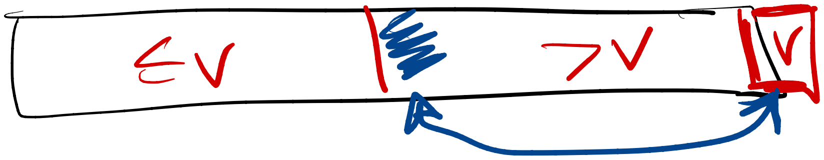


worst case p = extreme 0 or n-1

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

PARTITION (non-rec)





average case : $\text{Prop}(p = \text{particular position}) = \text{uniform} = \frac{1}{n}$

Split: $n-1 \mid 0$

$n-2 \mid 1$

$\vdots$

$0 \mid n-1$

$$E[T(n)] = \text{avg}(\text{all } T(n) \text{ in each case})$$

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

$p \to k$

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

// every T(k) appears twice

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

// T(0) rec call does not happen

$$T(n) = \frac{2}{n}\left(\sum_{k=1}^{n-1} T(k)\right) + n \qquad \boxed{nT(n) = 2\left(\sum_{k=1}^{n-1} T(k)\right) + n^2}$$

$n \to n-1$ // try for telescoping? $\sum$ o o o – o

$\sum$ o o o o o

$$T(n-1) = \frac{2}{n-1}\left(\sum_{k=1}^{n-2} T(k)\right) + n-1 \qquad \boxed{(n-1)T(n-1) = 2\left(\sum_{k=1}^{n-2} T(k)\right) + (n-1)^2}$$

⊖ first - second

$$nT(n) - (n-1)T(n-1) = 2[\cancel{T(1)} + \cancel{T(2)} \dots \cancel{} + T(n-1)] - 2[\cancel{T(1)} + \cancel{T(2)} \dots + \cancel{T(n-2)}] + n^2 - (n-1)^2$$

n^2+1-2n

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

$$nT(n) = (n+1)T(n-1) + 2n - 1 \qquad / \div n(n+1)$$

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

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

iterate $R(n) \leq R(n-1) + \frac{2}{n+1}$ (// smaller "$\leq$" because lost "-1")

$$R(n) \leq \left[R(n-2) + \frac{2}{n}\right] + \frac{2}{n+1}$$

$$\leq \left[R(n-3) + \frac{2}{n-1}\right] + \frac{2}{n} + \frac{2}{n+1}$$

general K $\leq R(n-K) + 2\left[\frac{1}{n+1} + \frac{1}{n} + \cdots + \frac{1}{n-K+2}\right]$

last $K \simeq n$ $\leq R(0) + 2\left[\underbrace{\frac{1}{n+1} + \frac{1}{n} + \cdots + \frac{1}{2}}_{\text{Harmonic} = \Theta(\log n)}\right]$

$$R(n) = \frac{T(n)}{n+1} = \Theta(\log n) \Rightarrow T(n) = (n+1)\,\Theta(\log n)$$

$$= \Theta(n \log n)$$

UB

LB (HW): best case $= \Theta(n \log n)$

QS

$\Downarrow$

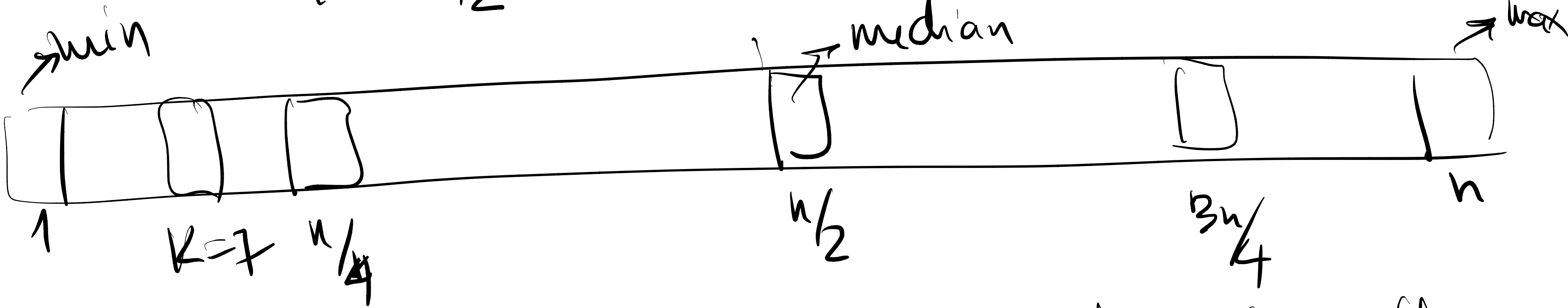
avg case $= \Omega(n \log n)$

QS-like Median Stats : Find item in array at rank K. (no sorting)

$K=1 \Rightarrow$ Find min

$K=n \Rightarrow$ find max

$K=n/2 \Rightarrow$ Find median (rank middle)

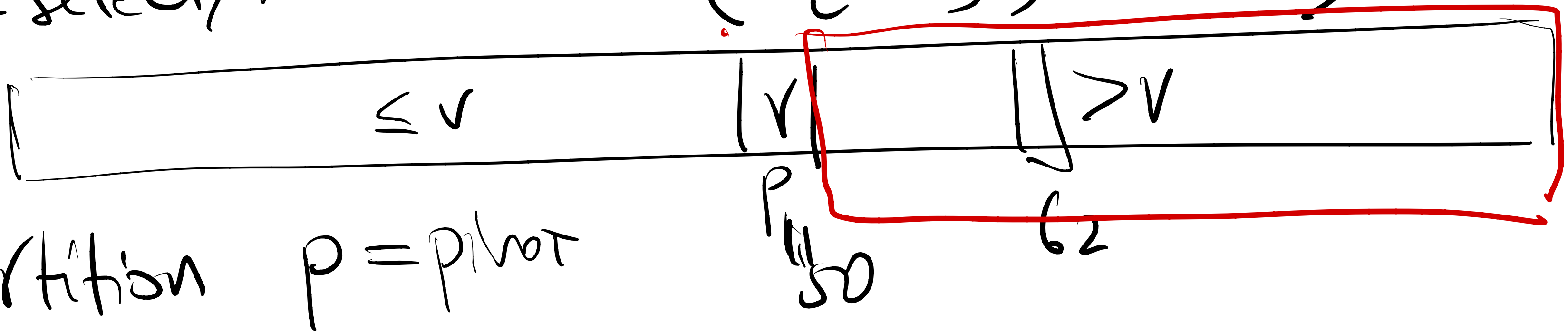


$K=3$: want the third-elem from the min

$K=n-2$: want the third-elem from the top

Solution (naive) : sort(A)
output A[k]

QuickSelect/Median Stat (A[b:e], rank k)



- partition p = pivot
- if $k == p$ done (output val at pivot, v)
- if $k < p$ QuickSelect(A[b:p-1], rank k)
- if $k > p$ QuickSelect(A[p+1:e], rank k-p?)

$T(n) = \underset{\text{part}}{n} +$ either $T(p)$ or $T(n-p-1)$

best case: $p = n/2$ $\quad T(n) = \underline{n} + T(\frac{n}{2})$ $\quad \Theta(n)$

worst case p = extreme $\underset{0 \text{ or } n-1}{}$ $\quad T(n) = n + T(n-1)$ $\quad \Theta(n^2)$

exercise

avg case $T(n) = \frac{1}{n}\left[\sum_{p=1}^{n-1} \max\begin{cases} T(p) \\ T(n-p-1) \end{cases}\right] + n$

pivot position prob

worst case rec

$\Theta(n)$

⊛

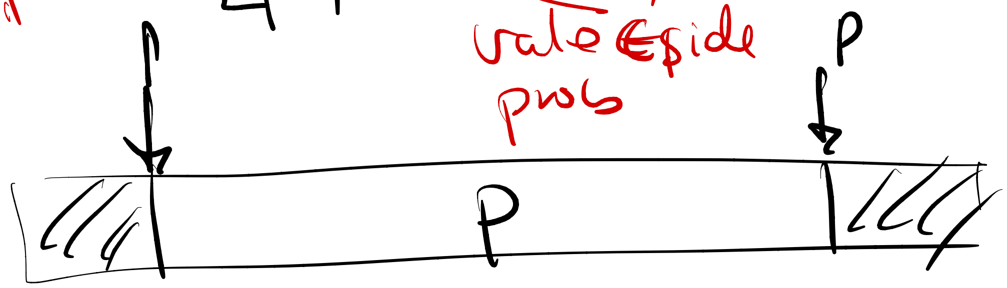
or weighted by probas inside

$$T(n) = \frac{1}{n}\left[\sum_{p=1}^{n-1} \frac{p}{n}T(p) + \frac{n-p-1}{n}T(n-p-1)\right] + n$$

pivot prob

value side prob

"normal-non-worst case"



$n \cdot r\% \leq P \leq n - nr\%$

$$T(n) = \boxed{n + T(n - nr\%)}$$

$$\boxed{\overset{c=1}{n} + \underset{a}{1} \cdot T\Big(n \cdot \underbrace{\frac{1-r}{100}}_{b}\Big)}$$

$\Theta(n)$

Theoretical Interest-only : eliminate the worst

find median

$$T(n) = \Theta(n) + \Theta(n) + T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right)$$

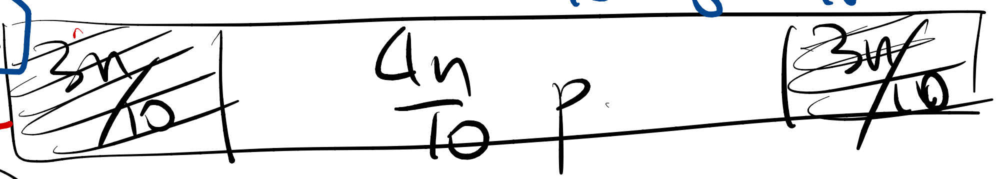
split, sort all groups, max-sider rec, of medians

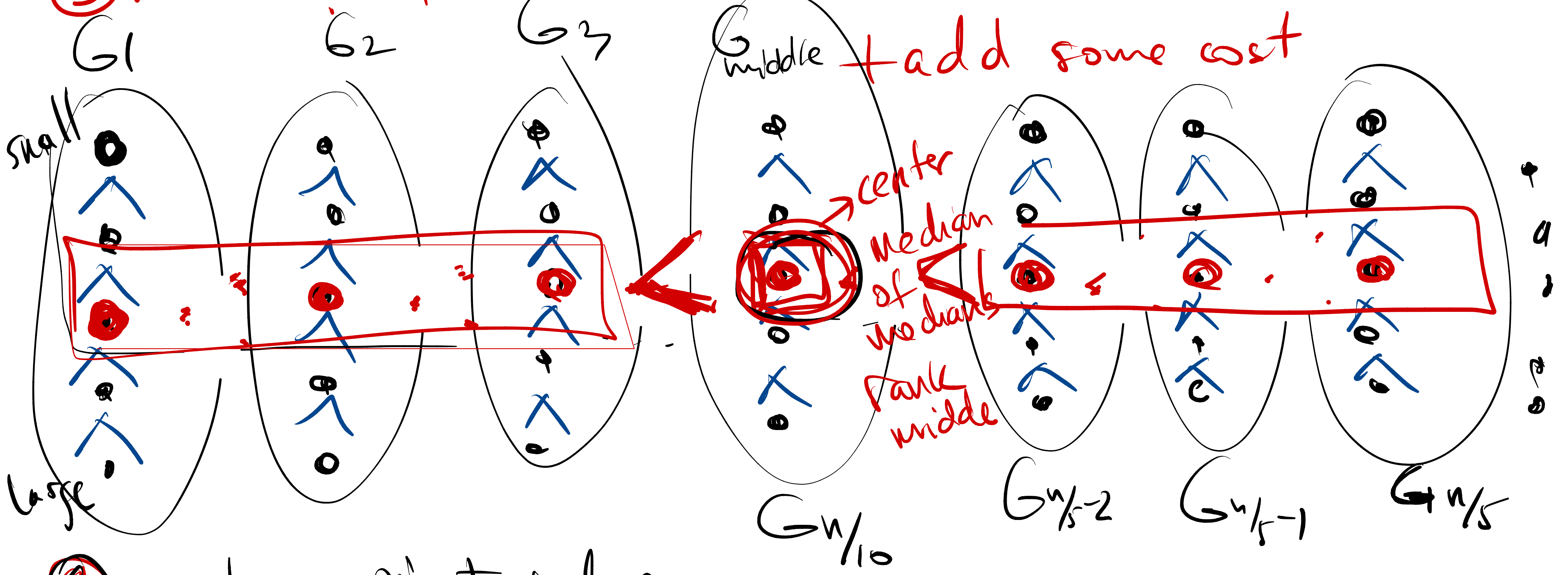
$$\frac{7}{10} + \frac{1}{5} = \frac{9}{10} < 1$$

① Split in groups of 5 — split

② Sort 5 elem in each group — sort all groups

③ median of median-of-group





④ center = pivot value

3 elem guaranteed smaller ← $\frac{3n}{10}$ groups left $\leq$ center $\leq$ $\frac{n}{10}$ × 3 elem per group, groups right

Ⓣⓗ : $\forall$ comparison-based sort works all the time $\Big\} \Rightarrow \Omega(n \log n)$

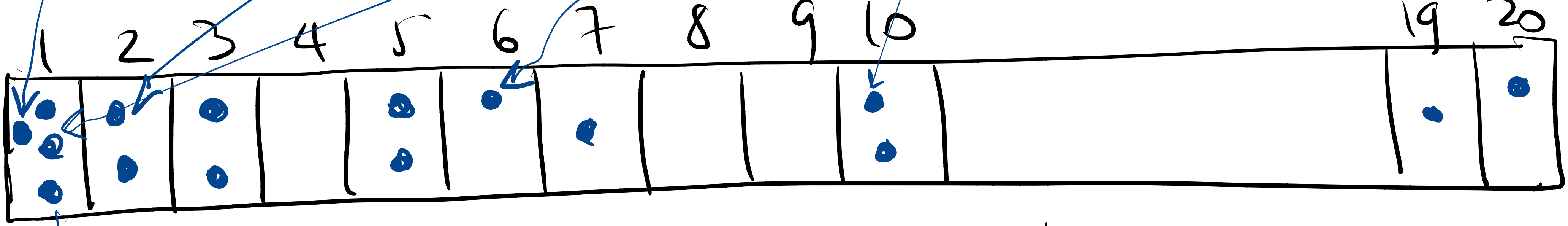
LINEAR-SORT (any values)

Counting Sort $A = [1, 3, 10, 2, 20, 5, 1, 6, 1, 3, 10, 19, 1, 5, 7, 2]$

Range of values [1 : 20] discrete

$|Range| = K = 20$

- create K buckets (one per possible value)



- traverse array count values → buckets
- print each bucket # times according to count

1 1 1 1 2 2 3 3 5 5 6 7 10 10 19 2

counting $\longrightarrow \Theta(n)$

buckets $\longrightarrow \Theta(K)$

printing buckets (x count)   output n values

for every bucket
    for # counts in bucket $\Theta(n+K)$

Ideal case: n large   K = small

population 7B   age (years) 1 : 200

Worst case   small n   large K

10 employees   salaries

Run Time   30k – 1M

Reasonable K = $\Theta(n) \longrightarrow \Theta(n)$

$|\text{Range}| = 970{,}000$

Radix Sort

	n values Counting sort for last digit	count sort middle digit (stable)	count sort first digit (stable)
329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

sorted? Yes if Count-sort stable!

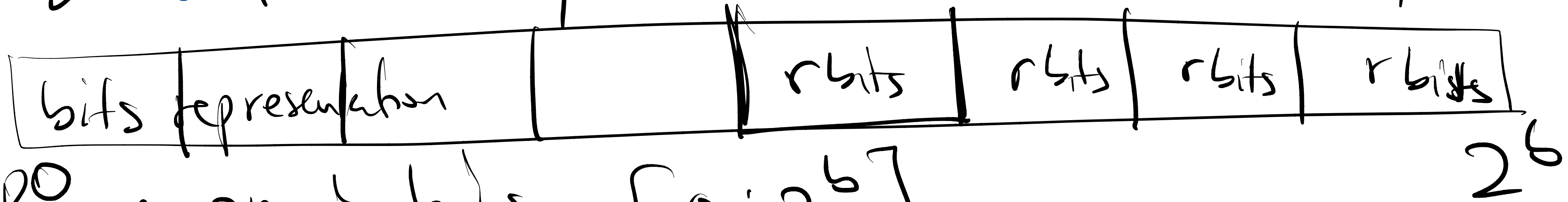
exercise

counting sort stable

Range [0:9]

$\Theta(n+10) \times$ #digits (Range 1:max) b = #bits in binary

b = fixed by task



Range on b bits $[0 : 2^b]$

pick digits/base #bits per digit = r

ex $r=3 \Rightarrow$ base $= 2^3 = 8$ base $= 2^r$

$$\boxed{r=4 \Rightarrow \text{base } 2^4 = 16}$$

$r=5 \Rightarrow$ base 32

$\#\text{digits} = \frac{b}{r}$ Range $[0 : 2^r-1]$ $|\text{Range}| = 2^r$

Radix Sort RT :

Counting Sort #digits

$$\boxed{\Theta(n+2^r) \times \frac{b}{r}}$$

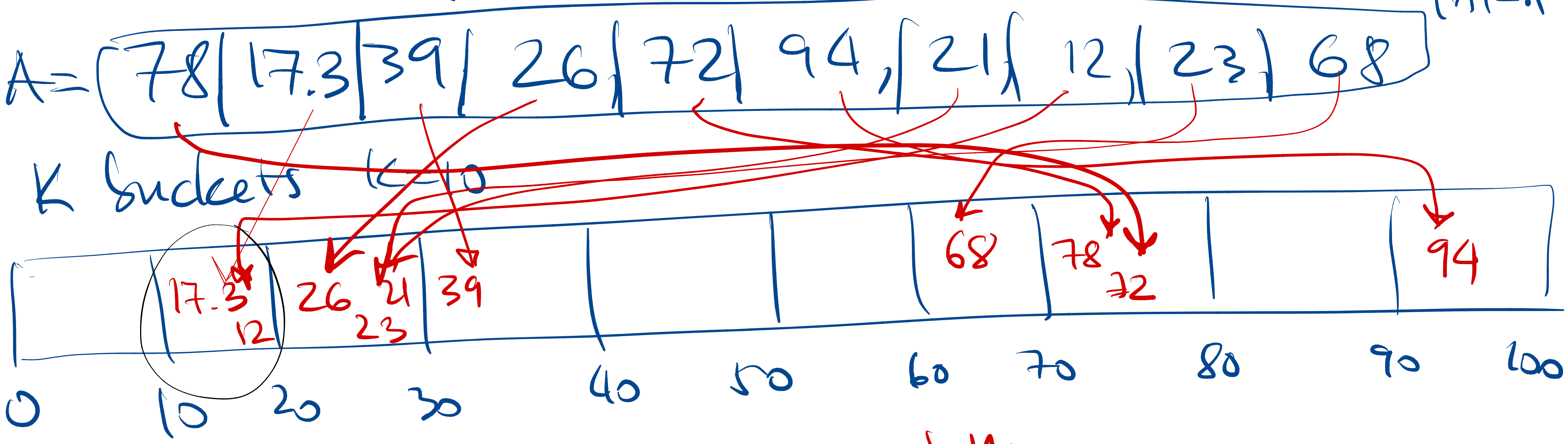
$b = ?$

max $n = 2^b$

$r \simeq \log n \Rightarrow RT = \Theta(bn/\log(n))$

Bucket sort ($\simeq$ Counting sort but bucket = range)

$|A| = n$



bucket
[0-10]

$n = n_1 + n_2 \ldots + n_K$ $E[n_i] = \frac{n}{k}$

elem n_i = # elem in bucket i

const $\Theta(1)$ per item

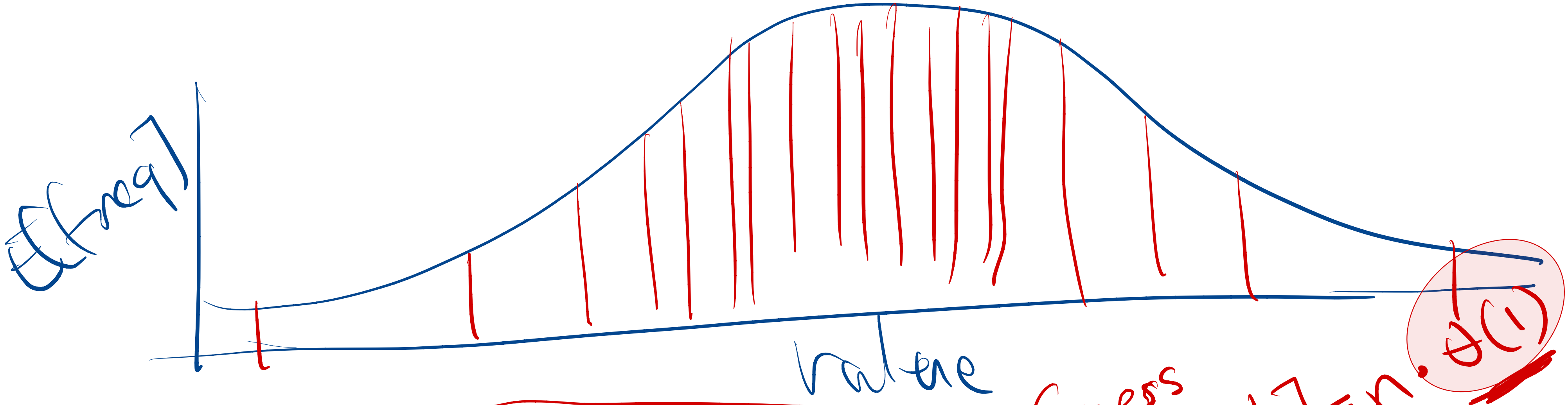
- "place" items into buckets (linear traversal) $\Theta(n)$
- sort each bucket (insertion sort) → ? $\sum_{i=1}^{K} \Theta(n_i^2)$
- output sorted values in $buck_1$, $buck_2$ . . . buck k

works!

K to use is $K = n$

work efficient

assume values distribute uniform over buck



$\boxed{k=n} \to \boxed{E[n_i] = \frac{n}{n} = 1}$

Guess

$E[\text{insert sort}] = \underline{n} \cdot \Theta(1)$

- placing → $\Theta(n)$
- insert sort → $\Theta(n)$
- printing $\Theta(n+k) = \Theta(n)$

} Linear

$n = n_1 + n_2 + \cdots$

$\downarrow$ # items in bucket1

n_k ~~$E[X^2] \neq (E[X])^2$~~

$\downarrow$ # items bucket k

$(k = n)$

$E[X] = 1$

$E[X^2] = ?$

$n_i = R.V.$

$$\boxed{E[n_i] = \frac{n}{k}} \overset{k=n}{=} 1$$

$$T(n) = \underline{\theta(n)} + \sum_{b=1}^{n} O(n_b^2)$$

$$E[T(n)] = \underline{\Theta(n)} + E\left[\sum_{b=1}^{n} O(n_b^2)\right] = \Theta(n) + \sum_{b=1}^{n} \boxed{E[n_b^2]}$$

$\boxed{\leq 2}$ const

$$\leq \Theta(n) + \Theta(n) = \Theta(n)$$

indicator R.V $X_{ij} = \begin{cases} 1 & \text{if item } j \to \text{bucket } i \\ 0 & \text{otherwise} \end{cases}$

i = fixed

n_i = # of items in bucket i $= \sum_{j=1}^{n} X_{ij}$ $\quad n_i^2 = \left(\sum X_{ij}\right)^2$

bucket 3 : $n_3 = \sum_{j=1}^{n} X_{3j}$

i = bucket
j = elem

$n_i = \sum_j x_{ij}$

$n_i^2 = \left(\sum_{j=1}^{n} x_{ij}\right)^2$

$\left(\left(\sum a_i + b + c\right)\right)^2$

$\sum a^2 + \sum ab$

$$E[n_i^2] = E\left[\left(\sum_j x_{ij}\right)^2\right] = E\left[\sum_j x_{ij}^2\right] + E\left[\sum_{j \neq k} x_{ij} \cdot x_{ik}\right]$$

$$= \sum_{j=1}^{n} E[x_{ij}^2] + \sum_j \sum_{k \neq j} E[x_{ij} \cdot x_{ik}]$$

$$E[x_{ij}^2] = 1 \cdot \underset{1/n}{P(x_{ij} = 1)} + 0 \cdot -$$

$x_{ij} \cdot x_{ik} = 1$: both j, k go to bucket i

indep $\quad 1/n^2 = \frac{1}{n} \cdot \frac{1}{n}$

$$= \sum_{j=1}^{n} \frac{1}{n} + \sum_j \sum_{k \neq j} \frac{1}{n^2}$$

$$\Theta(1) + \Theta(1) = \Theta(1)$$