CSE 3500

HW2 STAT

High: 100

Low: 22

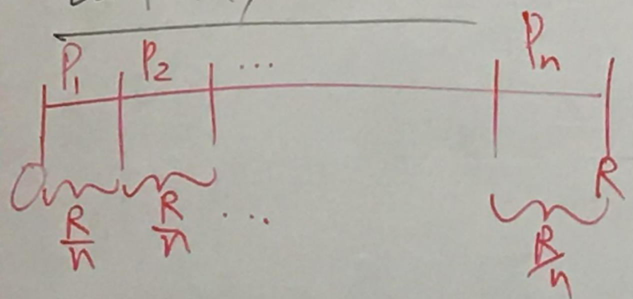
μ: 95

σ: 13

EXAM 2 on 11-15-16 @ 5PM IN MONT 104

(1) $X = k_1, k_2, \dots, k_n \in [0, R]$

SORT X;



(1) CREATE AN ARRAY $A[1:n]$ OF EMPTY LISTS;

(2) for $1 \le i \le n$ do

INSERT k_i into LIST $A\left[\left\lceil \frac{k_i}{R/n} \right\rceil\right]$;

(3) for $1 \le i \le n$ do

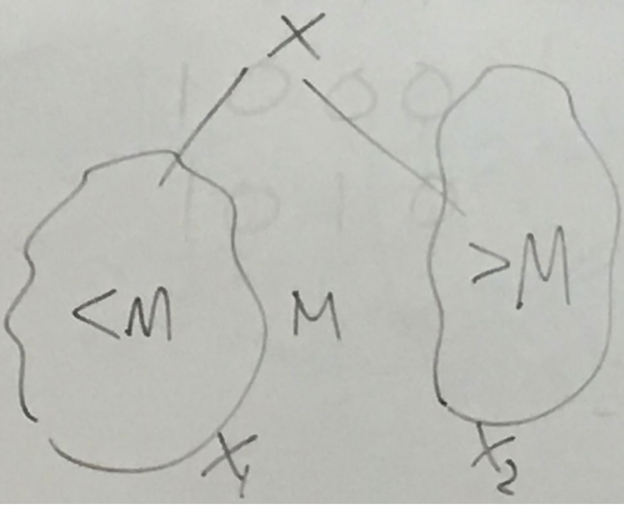
Sort and output the keys in the list $A[i]$;

Each list has an expected $O(1)$ keys & hence it takes an expected $O(1)$ TIME to each list;

$\Rightarrow$ The expected total run time is $O(n)$.

(2) $X = k_1, k_2, \ldots, k_n$.

FIND the median M of $X \Rightarrow O(n)$ TIME.



Case 1: $|X_1|$ is EVEN and $|X_2|$ is EVEN.

If so, output M and quit;

Case 2: $|X_1|$ is even and $|X_2|$ is ODD;

If so, Look for the Special element recursively in X_2;

Case 3: $|X_1|$ is odd and $|X_2|$ is EVEN.

If so, Look for the Special element in X_1 recursively.

Let $T(n)$ be the Run TIME.

$$T(n) = O(n) + n + T\left(\frac{n}{2}\right)$$
$$= T\left(\frac{n}{2}\right) + \theta(n).$$
$$= \theta(n).$$

(4) SORT the objects in NON decreasing order of Volumes.

Let the Sorted sequence of Volumes be $x_1, x_2, \ldots, x_n$

IDentify k s.t.

$$\sum_{i=1}^{k} x_i \leq m \text{ and } \sum_{i=1}^{k+1} x_i > m.$$

Output $x_1, x_2, \ldots, x_k$.

Run TIME $= O(n \log n)$.

PROOF OF OPTIMALITY:

let $g_1, g_2, \ldots, g_k$ be the greedy output.

let $O_1, O_2, \ldots, O_q$ be the OPTIMAL output.

If these two are not identical, then let j be the least index s.t.

$$g_j \neq O_j.$$

In this case $g_j < O_j$

5 0/1 KNAPSACK

where each $x_i \geq 0$

we have to find $x_1, x_2, \dots, x_n$ s.t.

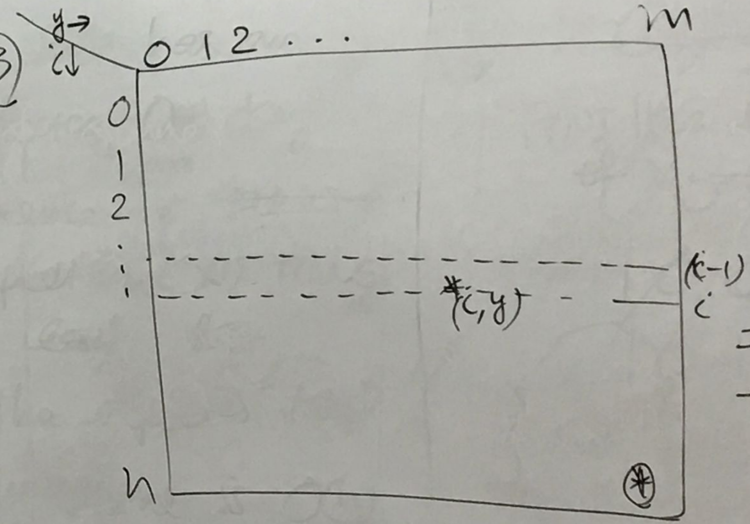
$$\sum_{i=1}^{n} w_i x_i \leq m \text{ and}$$

$$\sum_{i=1}^{n} p_i x_i \text{ is MAX.}$$

1 $f_i(y)$ = OPTIMAL PROFIT for objects 1 through i when the capacity is y.

2 $$f_i(y) = \mathrm{Max}\left\{ f_{i-1}(y),\ f_{i-1}(y-w_i)+p_i,\ f_{i-1}(y-2w_i)+2p_i,\ \dots,\ f_{i-1}(y-kw_i)+kp_i \right\}$$

$$k = \left\lfloor \frac{y}{w_i} \right\rfloor$$

3 

For every entry we spend $O(m)$ TIME.

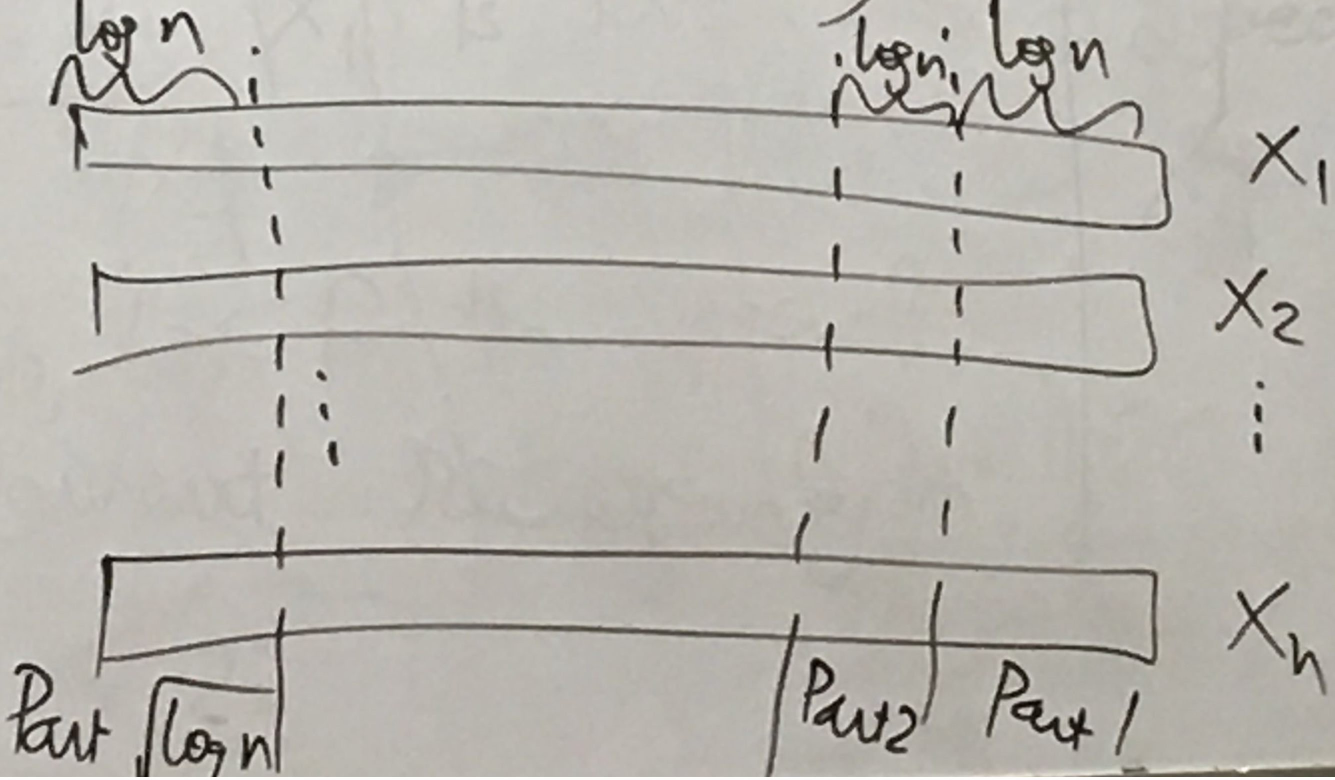
$\Rightarrow$ TOTAL RUN TIME $= O(m^2 n)$.

2014

1. INPUT: $k_1, k_2, \ldots, k_n$; $k_i \in [1, n^{\sqrt{\log n}}]$ INTEGER

Sort in $O(n\sqrt{\log n})$ TIME.

EACH INTEGER has $\sqrt{\log n}\,\log n$ BITS.



for $1 \le i \le \sqrt{\log n}$ do

Sort the integers w.r.to their Part i;

Each phase of sorting takes $O(n)$ TIME.

$\Rightarrow$ RUN TIME of the entire algorithm $= O(n\sqrt{\log n})$.

(2) INPUT:

$X = k_1, k_2, \ldots, k_n$; AN INTEGER $k \le n$.

Goal: Partition X into $X_1, X_2, \ldots, X_k$ s.t. $|X_i| = \frac{n}{k}$ and any element in X_j is greater than any element in X_{j-1} in $O(n \log k)$ TIME.

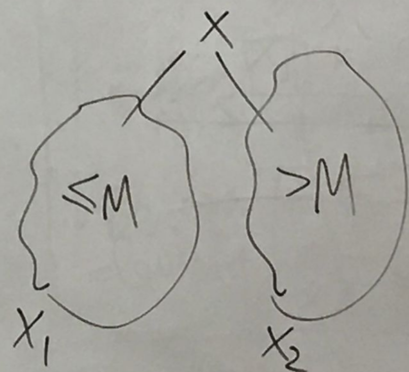
Ex. $X = 3, 11, 25, 18, 14, 6$

$k = 2$.

$X_1 = 3, 11, 6$; $X_2 = 25, 14, 18$.

(1) Find the median M of X in $O(n)$ TIME.

(2) Partition X into X_1 and X_2.



(3) Recursively partition X_1 into $k/2$ PARTS; output

Recursively partition X_2 into $k/2$ parts; output

(3) <u>INPUT:</u> $A_1, A_2, \dots, A_k$

EACH A_i IS $n \times n$;

INTEGERS $n_1, n_2, \dots, n_k$

<u>Output:</u> $A_1^{n_1} A_2^{n_2} \cdots A_k^{n_k}$

<u>Algorithm:</u>

(1) <u>for</u> $1 \leq i \leq k$ <u>do</u>

Compute $A_i^{n_i} = B_i$

using repeated squaring;

(2) Result $= B_1$;

<u>for</u> $2 \leq i \leq k$ <u>do</u>

Result = Result $\times B_i$

$A_i^{n_i}$ Can be computed in

$O\left(M(n) \log n_i\right)$ TIME.

$$N = \prod_{i=1}^{k} n_i$$

$\Rightarrow$ Step 1 takes

$$\sum_{i=1}^{k} O\left(M(n) \log n_i\right)$$

$$= O\left(M(n) \sum_{i=1}^{k} \log n_i\right)$$

$$= O\left(M(n) \log N\right) \text{ time}$$

Step 2 takes $O\left(k\, M(n)\right)$ time

Total Run time is

$$O\left(M(n) \log N\right).$$