CSE 532

HW - 3

RAHUL BHANSALI : 111401451

JATIN GARG : 111462753

Q.1.)

Assignment - 3

The total sectors for a given radius r = kr
↳ constant.

The total sectors for radius from .75 to 1.75 (TS) = $\int_{.75}^{1.75} kr\,dr$

$= \frac{k}{2} r^2 \Big|_{.75}^{1.75} = \frac{k}{2}(1.75^2 - .75^2)$

$= \frac{5k}{4}$. —— (1)

The probability that a sector from a Track i gets selected $= \frac{\#\text{sectors in track } i}{\#\text{total sectors}} = \frac{kr_i}{TS}$ [where r_i is radius of track i]

Similarly, probability that a sector from a Track j gets selected $= \frac{\#\text{sectors in track } j}{\#\text{total sectors}} = \frac{kr_j}{TS}$ [where r_j is a radius of track j].

∴ Expected ~~Value~~ number of tracks travelled when moving from a sector in Track i to a sector in Track j = Probability that a sector in track i & a sector in track j get selected × Tracks travelled from track i to track j

$$= \int_{0.75}^{1.75}\int_{0.75}^{1.75} \frac{k r_i}{TS} \times \frac{k r_j}{TS} \times |r_j - r_i| \, dr_i \, dr_j$$

Expanding the modulus, we have →

$$= \frac{k^2}{TS^2} \int_{0.75}^{1.75} \left[\int_{0.75}^{r_1} r_1 r_2 (r_1 - r_2) \, dr_2 + \int_{r_1}^{1.75} r_1 r_2 (r_2 - r_1) \, dr_2 \right] dr_1$$

= From (1) $TS = \frac{5k}{4}$ ⇒ $TS^2 = \frac{25}{16} k^2$ ⇒ $\frac{k^2}{TS^2} = \frac{16}{25}$

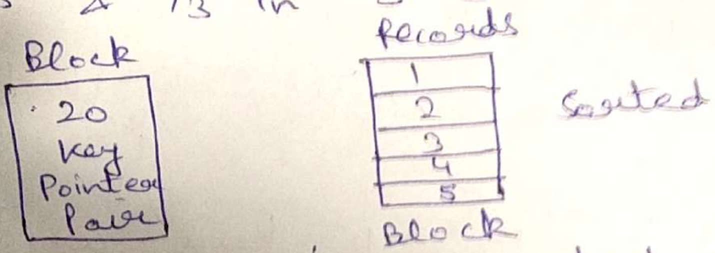
$$\Rightarrow \frac{16}{25} \int_{0.75}^{1.75} \left(\int_{0.75}^{r_1} r_1 r_2 (r_1 - r_2) \, dr_2 + \int_{r_1}^{1.75} r_1 r_2 (r_2 - r_1) \, dr_2 \right).$$

⇒ On evaluation, the integral comes out to be →

0.504167

$$\Rightarrow \frac{16}{25} \times 0.504167 = \underline{\underline{0.3226}}$$ [Ans].

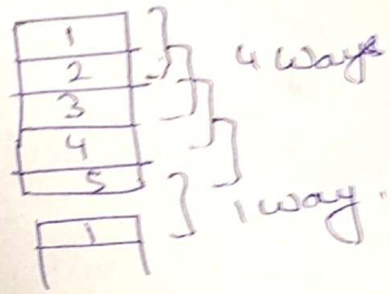
Q.2.) (a.) 1/3 keys appear in one record, 1/3 in 2 records & 1/3 in 3 records.



⇒ So lets see what is the probability that 2 records appear in same & different blocks.

Probability 2 records appear in same block = 4/5

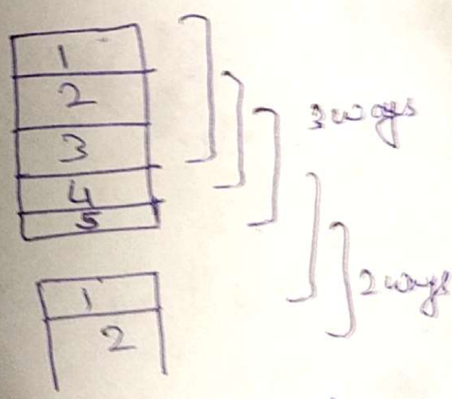
Probability 2 records appear in different blocks = 1/5



⇒ Now lets see what is the probability that 3 records appear in same & different blocks.

Probability 3 records appear in same block = 3/5

Probability 3 records appear in different blocks = 2/5



Now, we will calculate ~~pt~~ expected value of number of disk access.

We first bring key-pointer block in memory & then record block.

So here we would also consider whether 2 same key lie in same or different key pointer block.

Expected Value (Disk Access) =

① $\frac{1}{3}[2]$ ⇒ 1/3 keys which occur in 1 record would require 2 block access.

② $\frac{1}{3}\left[\frac{19}{20}\left(\frac{4}{5}\right)(2) + \frac{19}{20}\left(\frac{1}{5}\right)(3) + \frac{1}{20}\left(\frac{4}{5}\right)(3) + \frac{1}{20}\left(\frac{1}{5}\right)(4)\right]$

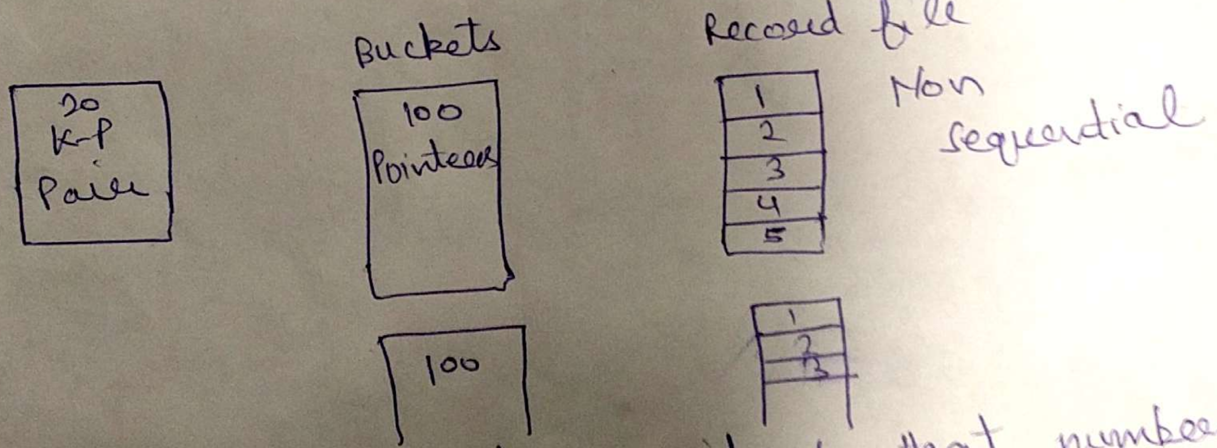
⇒ 1/3 keys which occur in 2 records.

③ $\frac{1}{3}\left[\frac{18}{20}\left(\frac{3}{5}\right)(2) + \frac{18}{20}\left(\frac{2}{5}\right)(3) + \frac{2}{20}\left(\frac{3}{5}\right)(3) + \frac{2}{20}\left(\frac{2}{5}\right)(4)\right]$

⇒ 1/3 keys which occur in 3 records.

① + ② + ③ ⇒ 2.25 Block Access.

(b.)



So for these we would consider that number of record blocks are very large & so the probability that 2 or 3 records lie in same bucket is 0. They all lie in different buckets.
Now, we would also consider probabilities that pointer for a given key lie in same or different blocks (records, pointer-key blocks &

For key in 1 record $= \frac{1}{3}[3]$

For key in 2 records $= \frac{1}{3}\left[\frac{99}{100}(4) + \frac{1}{100}(5)\right]$

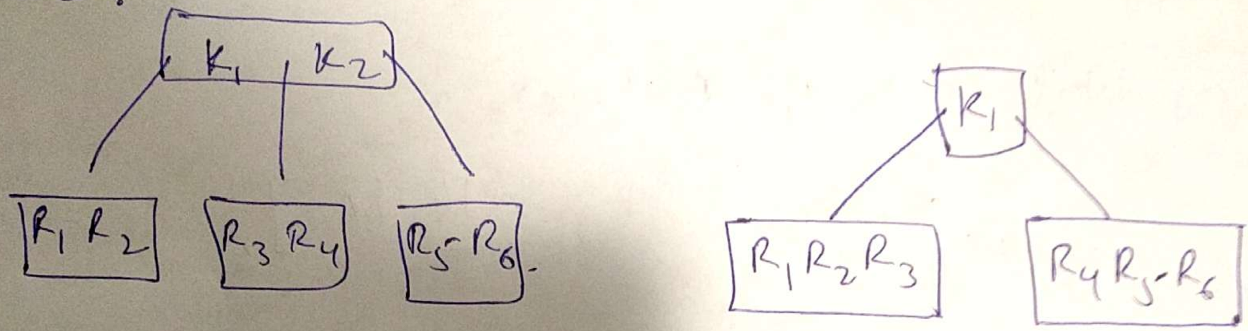
For key in 3 records $= \frac{1}{3}\left[\frac{98}{100}(5) + \frac{2}{100}(6)\right]$

Summing all 3. $\Rightarrow$ Expected value of disk access is 4.01.

Question 14.2.8 (3).

a) 6 ;

In this case there are a total of 2 different B-trees possible :

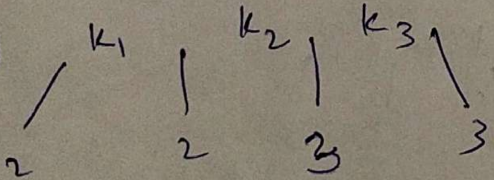


Where R_i = record i and K_i = key.

We can't permute each of the tree any further so the answer is $\underline{\underline{2}}$,

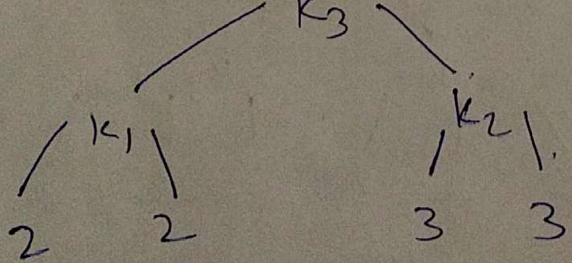
b) . 10 .

For this case the following trees are possible :

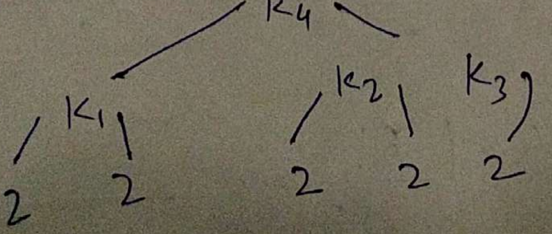
1) 

$\Rightarrow$ Total No. of permutations of trees for this case $= \frac{4!}{2!2!}$

where 2 and 3 denote the number of records in that leaf.

2) 

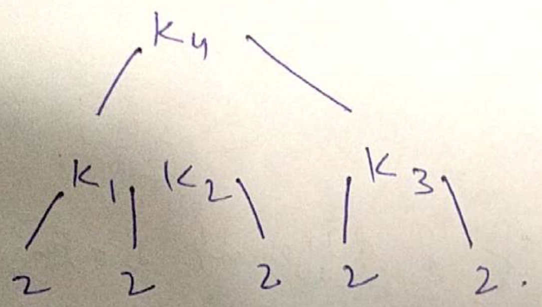
$\Rightarrow$ Total no. of permutations for this case $= \frac{4!}{2!2!}$

3) 

$\Rightarrow$ Permutations = 1.

Scanned by CamScanner

4.

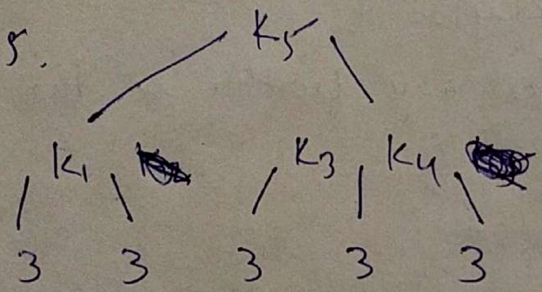


$\Rightarrow$ Total no. of permutations $= 1$.

Hence, total number of B-trees =

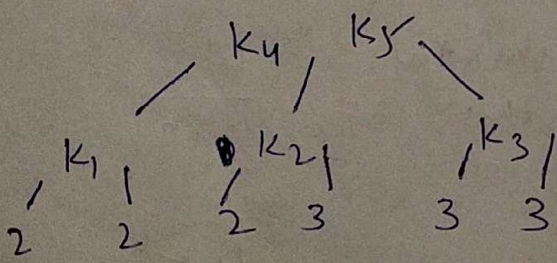
$$\frac{4!}{2!\,2!} \times 2 + 1 + 1 = \underline{\underline{14}} \text{ [Ans]}.$$

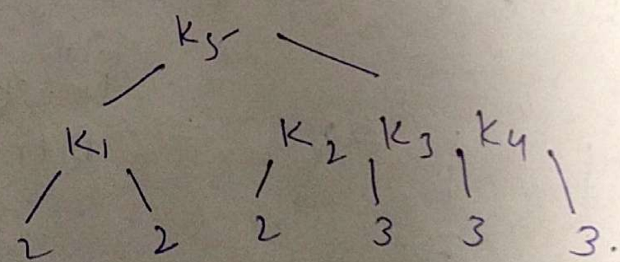
c) 15.

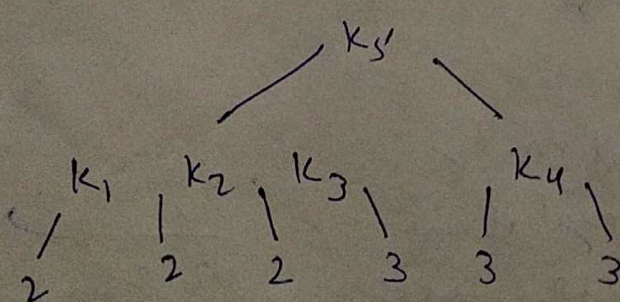


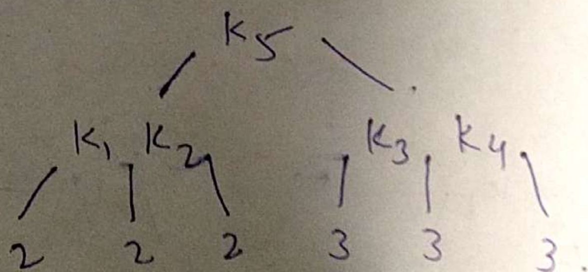
$\Rightarrow$ No. of permutations $= 2!$ —— (1)

—— x ——





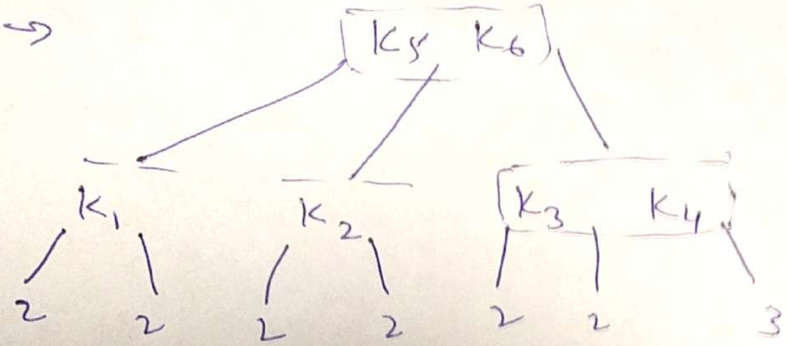




Total Permutation of each of the above four trees $= \dfrac{6!}{3!\,3!}$

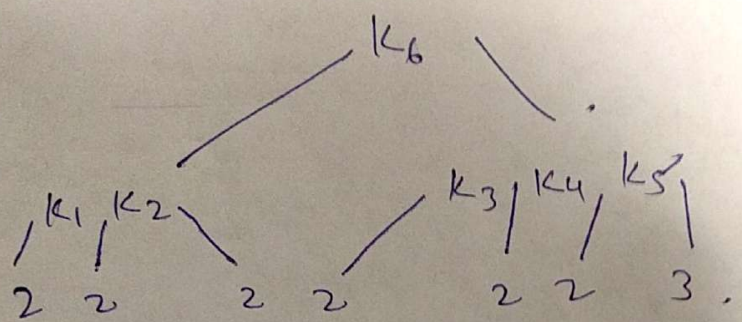
$\therefore$ Total trees ~~for the~~ $= 4 \times \left(\dfrac{6!}{3!\,3!}\right)$ —— (2)

We'll also have →



Total permutations of such a tree

$$= 3 \times \frac{7!}{6!} \quad -(3)$$



Total permutations of such a tree =

$$2 \times \frac{7!}{6!} \quad -(4)$$

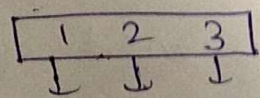
Hence, total trees = (1) + (2) + (3) + (4)

$$= 2 + 4 \times \frac{6!}{3!\,3!} + (3+2)\frac{7!}{6!} = \underline{\underline{117}} \text{ [Ans]}.$$

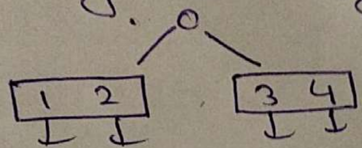
14.2.9.) Question : 4.)

It is given that we split a leaf node in 2 & 2. & a non leaf node in 3 & 2.

So we start with 1,2,3 as leaf node.

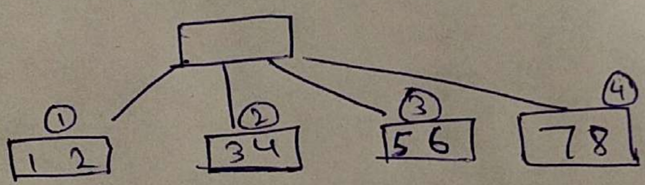


Now after adding next key 4, it splits like this:



So, we see that after filling a node with 3 & then adding next key would result in split.

2nd Level



Ist Level

So, 9 would get added in above 4th node & when 10 would come, it would result in split at 2nd level and formation of 3rd levels.

So 9 keys were accomadated at 2nd level & 10th result in formation of 3rd level.

As this is 3 key & 4 pointer B-Tree, ~~it~~ and we can see that 2 key are coming in each node & as there are 4 pointers, 4 such child nodes can be formed accomadating 2 keys each. But last ~~key~~ child node can accomodate 1 more key & addition of next key would split its parent node.

So, this logic ~~of~~ division is transferred from child to above levels forming a geometric progression.

2nd level is formed at $\Rightarrow 3^1 + 1 = 4^{th}$ node

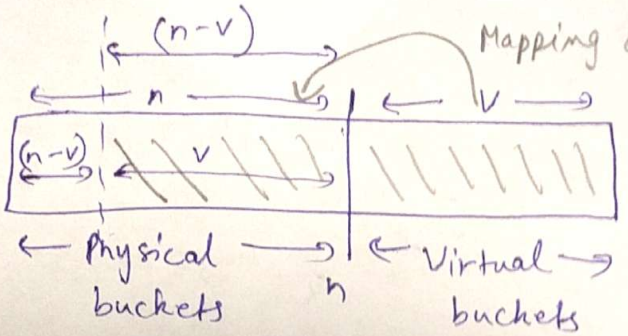
3rd level is formed at $\Rightarrow 3^2 + 1 = 10^{th}$ node.

4th level is formed at $\Rightarrow 3^3 + 1 = 28^{th}$ node.

5th level is formed at $\Rightarrow 3^4 + 1 = 82$ node

& so on.

5 (a)



Mapping of virtual bucket to physical bucket.

r = total records

k = records per block.

Let v be the number of virtual buckets

Thus, total buckets $= n + v = T$

Now, we use $\log_2 n$ bits for the keys $\therefore$ total buckets $=$

$2^{\lceil \log_2 n \rceil} = T$

Since, each key occurs its expected number of times (given) $\Rightarrow$

keys are distributed evenly (uniformly).

$\therefore$ keys mapped per bucket $= \dfrac{r}{T}$

$\therefore$ keys mapped for the first $(n-v)$ buckets $= \dfrac{r}{T}(n-v)$ — (1)

keys mapped for the next $2v$ buckets $= \dfrac{r}{T}(2v)$ — (2)

$\therefore$ For (1), No. of blocks required $= \left\lceil \dfrac{\frac{r}{T}(n-v)}{k} \right\rceil$ — (3)

For (2),

No. of blocks required $= \left\lceil \dfrac{\frac{r}{T}(2v)}{k} \right\rceil$ — (4)

$\therefore$ Total blocks required $=$

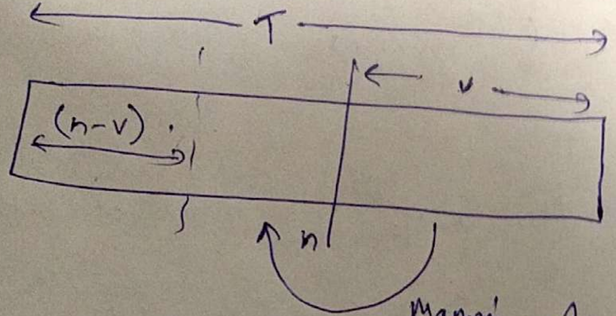
$\left\lceil \dfrac{r(n-v)}{Tk} \right\rceil + \left\lceil \dfrac{2rv}{Tk} \right\rceil \Rightarrow$

Scanned by CamScanner

$$\left\lceil \frac{cn \quad 2n - 2^{\lceil \log_2 n \rceil}}{2^{\lceil \log_2 n \rceil}} \right\rceil + \left\lceil \frac{2cn\left(2^{\lceil \log_2 n \rceil} - n\right)}{2^{\lceil \log_2 n \rceil}} \right\rceil$$

[Ans]

b).



Mapping of virtual buckets to physical.

$T = 2^{\lceil \log_2 n \rceil}$

This is similar to (a) except that now the number of keys mapped to a bucket is Poisson distributed.

Let X_i represents the number of records that map to bucket i.

$\therefore \; X_1 + X_2 + \dots + X_T = r \quad \text{—(1)}$

For first ~~(n-m)~~ $(n-v)$ buckets, we have the following:

expected No. of records in those $(n-v)$ buckets $\Rightarrow$

$$E[X_1 + X_2 + \dots + X_{n-v}] = E[X_1] + E[X_2] + \dots + E[X_{n-v}]$$

$\Rightarrow (n-v)\cdot\lambda \quad \text{—(1)}$ [Since expected value ~~of~~ from Poisson distribution $= \lambda$].

Similarly for the remaining $2v$ buckets, no. of records $=$

$$E\left[X_{n-v+1} + \dots + X_T\right] = E[X_{n-v+1}] + \dots + E[X_T].$$

$\Rightarrow (2v)\lambda$ [Expected value of Poisson distribution $= \lambda$]

$- (2)$.

From (1) $\rightarrow$ No. of blocks required $= \left\lceil \frac{(n-v)\lambda}{k} \right\rceil$

From (2) $\rightarrow$ No. of blocks required $= \left\lceil \frac{(2v)\lambda}{k} \right\rceil$

$\therefore$ Total blocks required =

$$\left\lceil \frac{(n-v)\lambda}{k} \right\rceil + \left\lceil \frac{(2v)\lambda}{k} \right\rceil$$

$$\Rightarrow \left\lceil \frac{(2n - 2^{\lceil \log n \rceil})\lambda}{k} \right\rceil + \left\lceil \frac{2\cdot(2^{\lceil \log n \rceil} - n)\lambda}{k} \right\rceil$$

[Ans].

Question 6:)

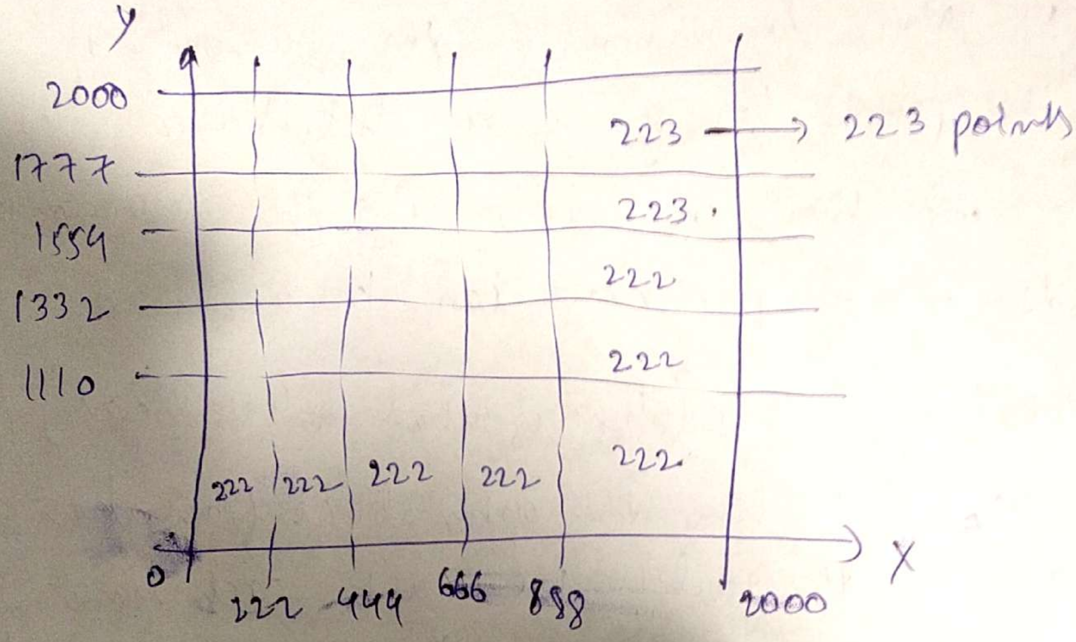
6. Basically, we want to maximize the number of grids where the points for $y = x$ is distributed, as only then the objective of minimizing the maximum number of grids in a partition can be acheived.

We have a total of 8 lines (4 horizontal & 4 vertical) that we can use to achieve the optimal configuration. How do we achieve the goal? We want each line to be drawn such that there's no "wastage" of grid and "new points" get inserted in the new grids being created ~~creating~~ by drawing a new line everytime.

For a total of 8 lines, we at max ~~basically~~ have 9 grids that can be filled ~~at max since~~ for the $y = x$ line. Thinking of it another way, each line (horizontal or vertical) cuts the $y = x$ line into two. So 8 lines will create 9 partitions of the $y = x$ line. Now, we can ~~distribute the points~~ create grids such that the maximum number of points in any grid is $\lceil n/9 \rceil$ (for a $n \times n$ grid)

here $n = 2000 \Rightarrow \lceil n/9 \rceil = 223$ [Ans]

The configuration can be achieved as follows →



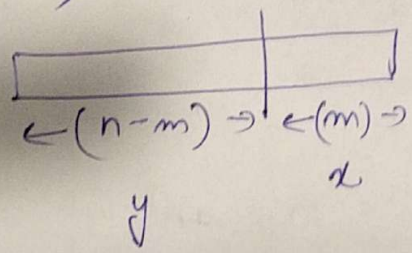
In general, for the $y = x$ line, for an $(n \times n)$ grid and if we have a total of (m horizontal & m vertical lines)

the formula becomes $\rightarrow \left\lceil \frac{n}{2m+1} \right\rceil$

Question - 7.)

14.5.6

(a)



If a query specified x, then we would have to look at all the ~~elements~~ buckets from the remaining $(n-m)$ bits i.e. 2^{n-m} ~~elements~~ buckets.

If a query specified y, then we would have to look at all the ~~elements~~ buckets from the remaining m bits i.e. 2^{m} elements.

Now, with probability p, a query for x comes →

∴ Expected no. of buckets to be examined when a query for x comes $= p \cdot 2^{n-m}$. (by definition of expectation).

Expected no. of buckets to be examined when a query for y comes $= (1-p) 2^{m}$.

∴ Total expectation $= p \cdot 2^{n-m} + (1-p) 2^{m}$ [Ans].

b) From previous part →

$$E = p \cdot 2^{n-m} + (1-p)2^m$$

To minimize E w.r.t. m → $\frac{dE}{dm} = 0$. ~~and w~~

$$\therefore \frac{dE}{dm} = -p \cdot 2^n \cdot 2^{-m} \cdot \ln 2 + (1-p)\, 2^m \ln 2 ,$$

$$\therefore 0 = -p \cdot 2^n \cdot 2^{-m} \ln 2 + (1-p)\, 2^m \ln 2 .$$

$$\Rightarrow (1-p)\, 2^m \cdot \cancel{\ln 2} = p \cdot 2^n \cdot 2^{-m} \cancel{\ln 2} ,$$

$$\Rightarrow 2^{2m} = \left(\frac{p}{1-p}\right) 2^n .$$

$$\Rightarrow 2m = \log_2 \left(\left(\frac{p}{1-p}\right) \cdot 2^n \right)$$ [Taking $\log_2$ on both sides]

$$\Rightarrow m = \frac{\log_2\left(\frac{p}{1-p}\right) + n}{2} \rightarrow \star$$

Now to verify its a minima point, we take the double derivative →

$$\frac{d^2E}{dm^2} = +\underbrace{p \cdot 2^n \cdot 2^{-m} \ln^2 2}_{+ve} + \underbrace{(1-p)}_{+ve} 2^m \ln^2 2$$

as we can see above expression is positive at

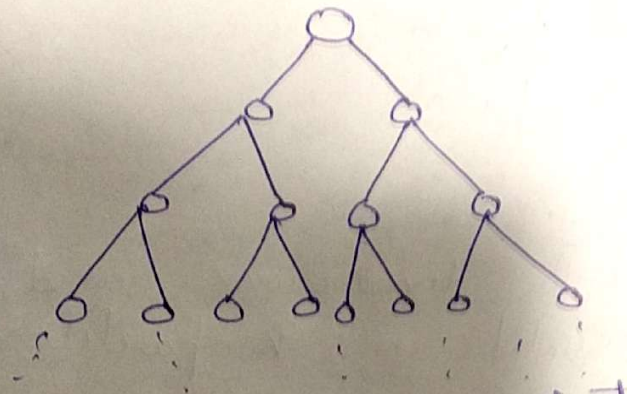
so $m = \frac{\log_2\left(\frac{p}{1-p}\right) + n}{2}$

So its a minima point.

Hence $m = \frac{\log_2\left(\frac{p}{1-p}\right) + n}{2}$ [Ans].

Q.8) kd tree is different from binary tree in that instead of making decision at every level which searching for a key we make decision at ~~exe~~ m dimensions out of every d dimensions.

Part a) $d=2$ & $m=1$



So, if we know root attribute we explore only 1 of its child & then we can not make decision at next level, so we have to explore both its children. So the series would be "(of nodes explored)

$\Rightarrow 2^0 \; 2^1 \; 2^1 \; 2^2 \; 2^2 \ldots$

Alternately we don't know root attribute & know next level attributes. So series would be:

$2^0 \; 2^0 \; 2^1 \; 2^1 \ldots$

So we will assume that for large height this sequence would be same. One will be floor & other will be ceiling as in:

No. of nodes explored $= 2^{\frac{h}{2}}$

where h is level of kd tree.

And h would be $\log_2 n$ where n is number of nodes at leaf level.

$\Rightarrow \approx 2^{\frac{\log_2 n}{2}} \Rightarrow 2^{\log_2 n^{1/2}} = \sqrt{n}$

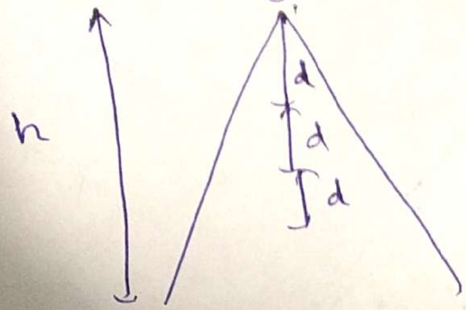
Scanned by CamScanner

Part b) d = Total dimensions

m = Known dimensions.

n = Total number of leaf-nodes

Total Level = $\log_2 n$



As d are total dimensions and we can divide total height in k partitions

where $k = \frac{\log_2 n}{d} = \frac{h}{d}$

& No. of child explored at each d^{th} level would be $1, 2^{d-m}, 2^{2(d-m)}$ & so on upto k^{th} partition.

So, formula for leaf explored

$$= 2^{(d-m)\left[h/d\right]} = 2^{k(d-m)}$$

$$= 2^{(d-m)\left[\frac{\log_2 n}{d}\right]}$$

Ans ⇒ $$2^{(d-m)\left[\frac{\log_2 n}{d}\right]}$$

Part c) To compare complexity of partitioned hash with kd tree, we are taking n records.

There are total h hash functions.

So total bits required to represent any record is hb where b is bits in each function.

Now in order to compare complexity with kd tree, we say that out of h hash functions we know m of them.

So, now total number of bits or attributes that we have to explore would be

$$2^{(h-m)b} = \text{Number of buckets we explore.} \quad \text{①}$$

There are n records represented by hb bits. So,

$$n = 2^{hb}$$

$$\log_2 n = hb$$

$$\frac{\log_2 n}{h} = b \quad \text{②}$$

Taking ① & ② we get:

$$\Rightarrow 2^{(h-m)\left[\frac{\log_2 n}{h}\right]}$$

where h is total dimensions equal to d in previous part.

So replacing h with d.

$$2^{(d-m)\left[\frac{\log_2 n}{d}\right]}$$

So, the complexity is same for both.

Question: 9

9. We basically need to figure out, given a random point, how many regions is it contained in i.e. given a random point in a region or node how many "children" of this region will we traverse.

Let area of root = X.

$\therefore$ Total area of subregions of root (i.e. children of root) = $1.5X$. (given)

$\therefore$ Average area of a subregion = $\frac{1.5X}{100}$ [as 100 subregions exists].

$\therefore$ Probability that a point lies in a subregion = $\frac{\text{area of a subregion}}{\text{total area}}$

$$= \frac{\frac{1.5X}{100}}{X} = \frac{1.5}{100} \quad \text{—(1)}$$

$\therefore$ Expected sub-regions that a point lies in = Probability of a point lying in a subregion $\times$ No. of subregions.

[by definition of Expectation] & using (1)

$$= \frac{1.5}{100} \times 100 = 1.5$$

So, ~~the~~ at the ~~first~~ second level we'll traverse 1.5 regions $\Rightarrow$ 1.5 blocks.

Similarly, at the next (third) level, we'll traverse →
(1.5 × 1.5) regions.

⇒ $(1.5)^2$ regions. ⇒ 2.25 blocks

Thus, total ~~regions~~ blocks traversed =

$$\underset{\substack{\text{(at root)}\\ \text{1st Level}}}{1} + \underset{\text{(2nd Level)}}{1.5} + \underset{\text{(3rd Level)}}{(1.5)^2} = \underline{4.75 \text{ blocks.}}$$

[Ans]

[There are total <u>3 levels</u>, as the R-tree will be like this →].