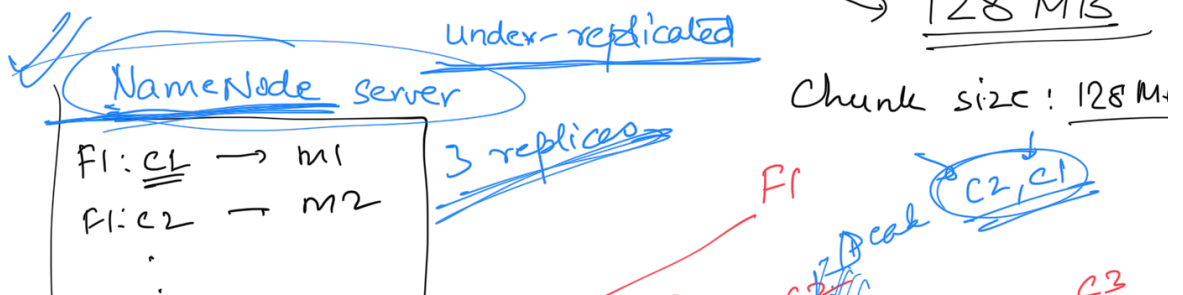# LARGE FILE STORAGE + PLACES
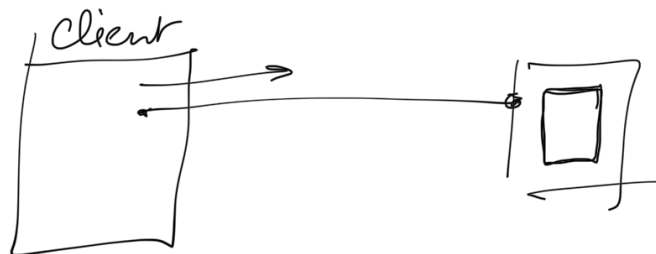
1. HOW TO STORE REALLY LARGE FILES?
2. HOW TO FIND NEAREST NEIGHBORS?

50TB file

1. Store it
2. Reliable
3. Download part of the
4. Analytics

500GB Chunk1

500GB Chunk2 . . . . . —

1MB
C1   C2   C3   C4 . . . . . — — 10,000

Collating

Cost to table
↳ too entri

F1: C1 ——→ m1
F1: C2 ——→ m2   Entries
F1: C3 ——→ m3
⋮

In memory
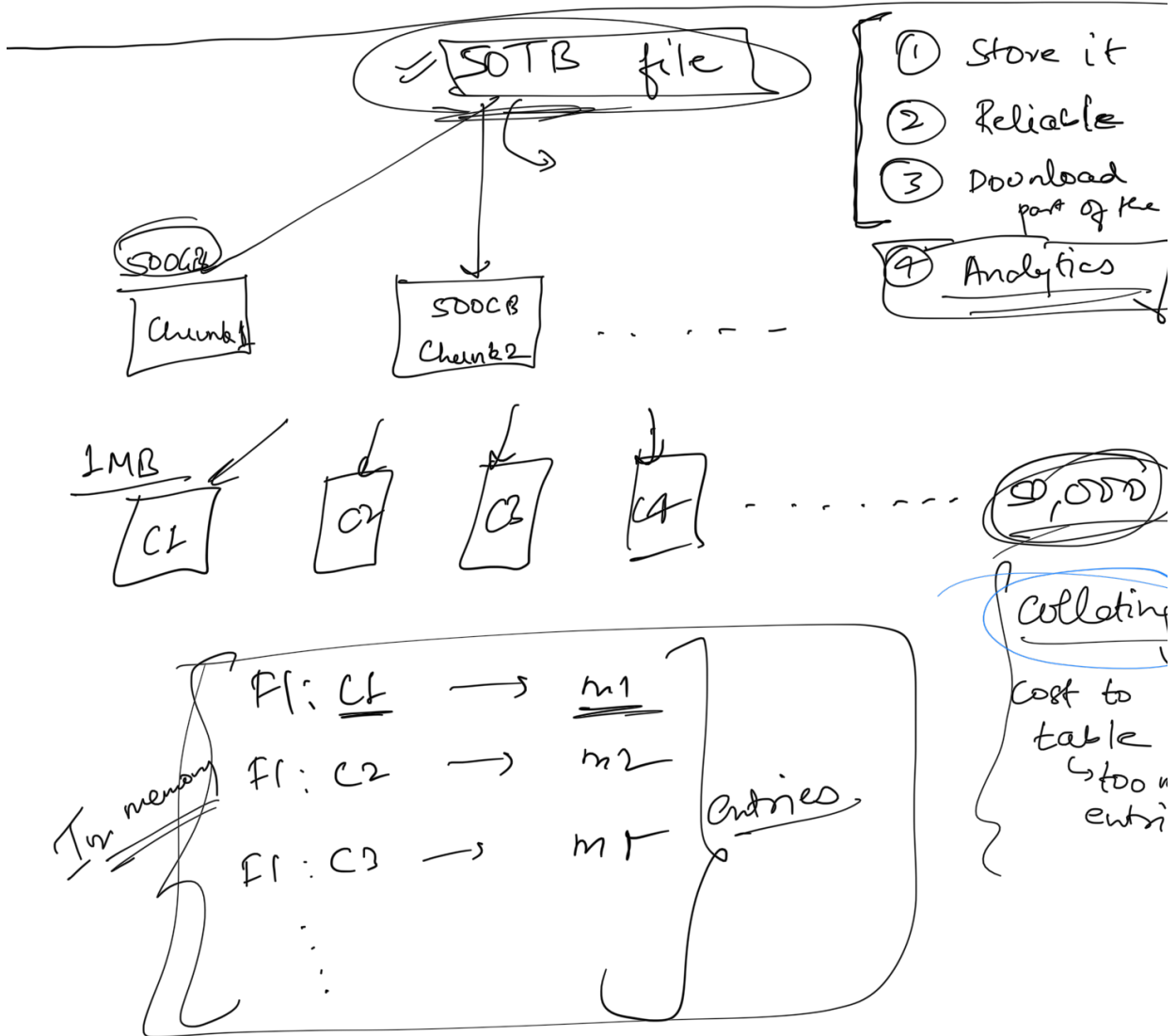
HDFS → | Hadoop Distributed File System
                                    ↳ 128 MB

Under-replicated

NameNode server

F1: C1 → m1   3 replicas
F1: C2 → m2
⋮

Chunk size: 128 M

F1

C2, C1

C3

C1   C3
C1   C2   C2   C2   C2
C2

DataNode

Rack-aware

Power

3 Power

Router
Switch

R1   R2   R3

Client — stream of bytes → Appserver → HDFS Client → DataNodes

streamde

NN

F1 C10   F1, C10

128 MB

Client

30 MB

Laptop 3GB → 30MB

part 1 — 30MB

part 2

Google Drive

10 parts

128

Datanodes are active

Datanodes

XN

→ Chunks

Chunk id → m1 m2, m3

C1
C2
C3

Files

assign intelligently

200 MB

100 MB

100 MB

128 MB

128M

72 MB

F1 → chunk

→ Chunk → machine mapping

Namenode does not require sharding

Datacentre

25 PB

128 MB

$$\frac{25 * 10^7 \text{ MB}}{128 \quad \text{MB}} \approx 2 * 10^8$$
$$\frac{}{100}$$

0.2 GB

$$\frac{25 \text{ PB}}{128 \text{ MB}} = \frac{25 * 1000}{}$$

$25 * 10^6$ GB

$25 * 10^9$ MB

Appservers

HDPS

datanodes

NameNode Server

→ replicas

F1 | Chunk
Ch | machine

F2   C3   C
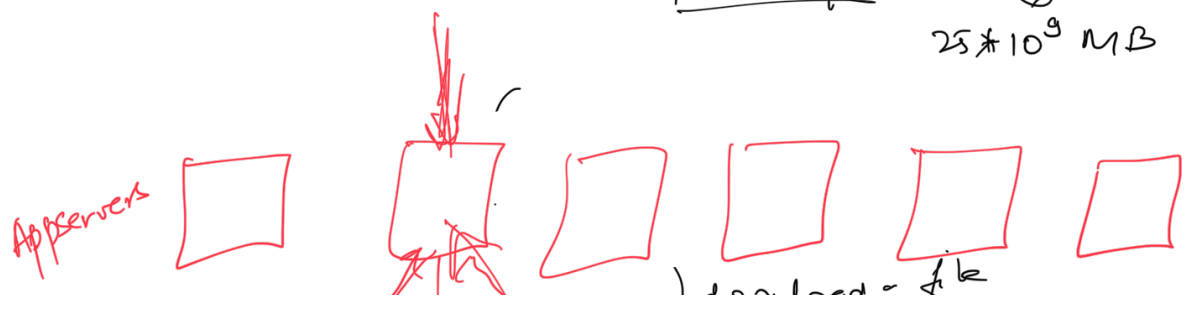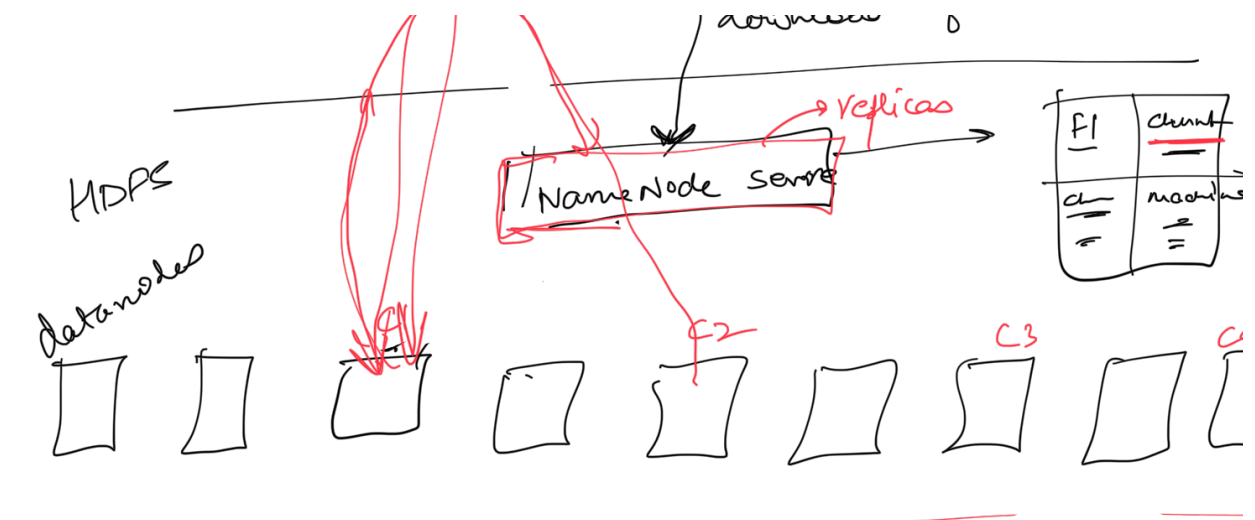
PLACES

① Nearest X restaurants / hotels / businesses.

② Places that are less than X kms me.

① (lat, long)

log-2
lat-2   lat+2
(lat, long)
log+2

SELECT * FROM restaurants
where latitude > lat - 2 AND l_{on} < lat+2
or lon > log - 2

→ 1 million

0.01 Latitude
      log

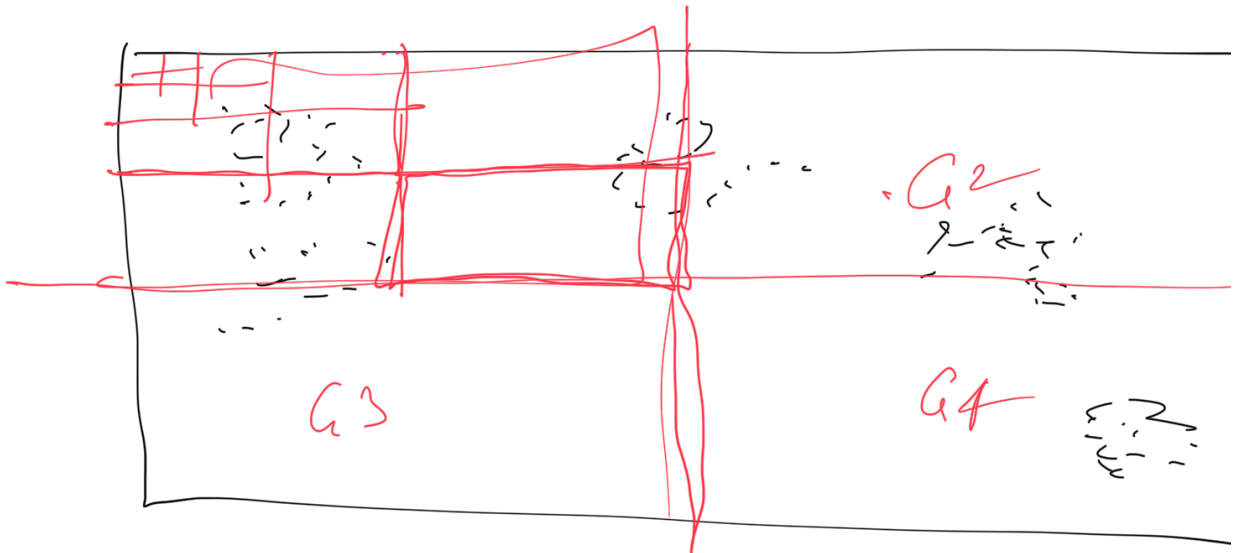① precision error ✓
② perf issue ✓

$\dot{\theta}$ (lat, long)

③ Choosing ⊗ ? ⊗

②
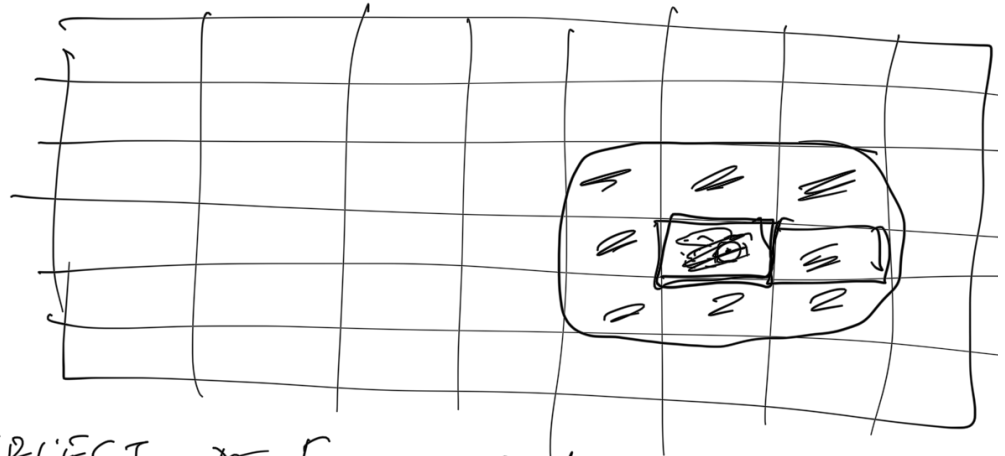
SELECT ⊗ From resta index
WHERE grid-id IN (100, 101, 102
......)

Indiron

50m
← 50m →

Village

2km
← 2km →

G2

G3   G4
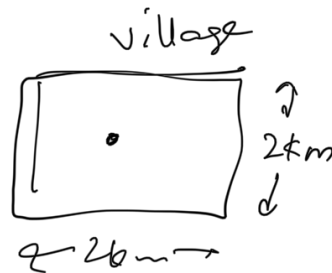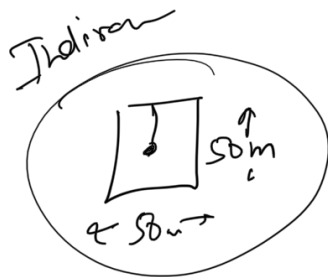
50 places   1 billion resta

(r,c)

Roof → entire world

LIST size > 50

Left side diagrams and annotations:

coord = ? c1-places
C1 size >50

c2 · c3 c4

C11 C1 C12 C14

r1 C1 ———— C2
r2

$\frac{(r1+r2)}{2}$ r1 C1 $\frac{C1+C2}{2}$ C2

$\frac{r1+r2}{2}$ r2

(r, c)

$r < \frac{r1+r2}{2}$ & $c < \frac{C1+C2}{2}$

c1 r1   r2

c2

Right side pseudocode:

process-node :

  if size <= 50 :
    return

  split into 4 parts
    c1, c2, c3, c4
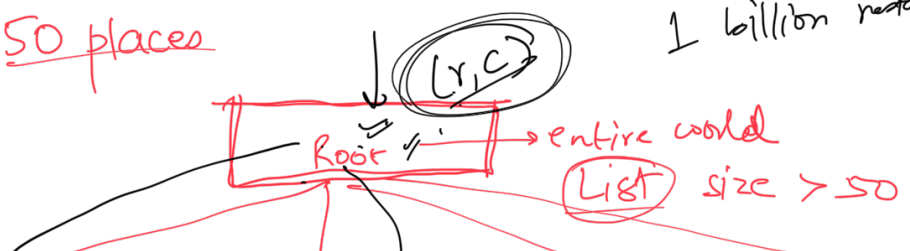
  process-node(c1)
  process-node(c2)
    :
    :

  return

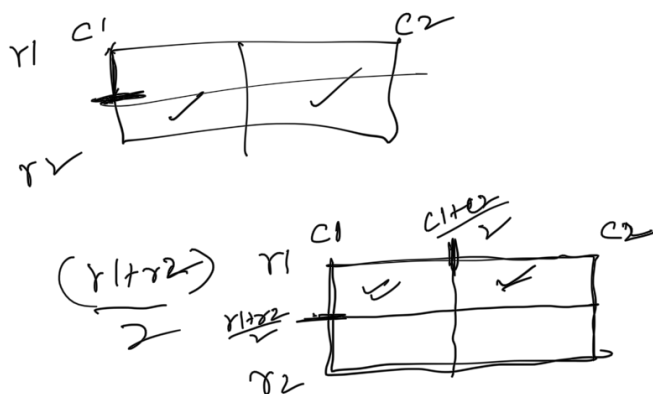add-place(root, coord):

  if root is a leaf
    root.places.
      (coord)
    if root.place
      > 50 :
      split into
        4 c
      return

  find which child co...
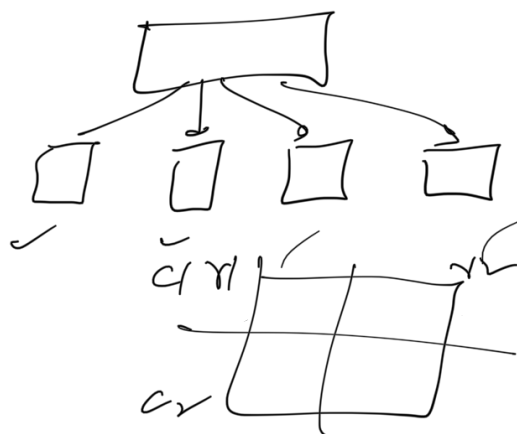   belongs to &
  add-place(child, coo...
  return

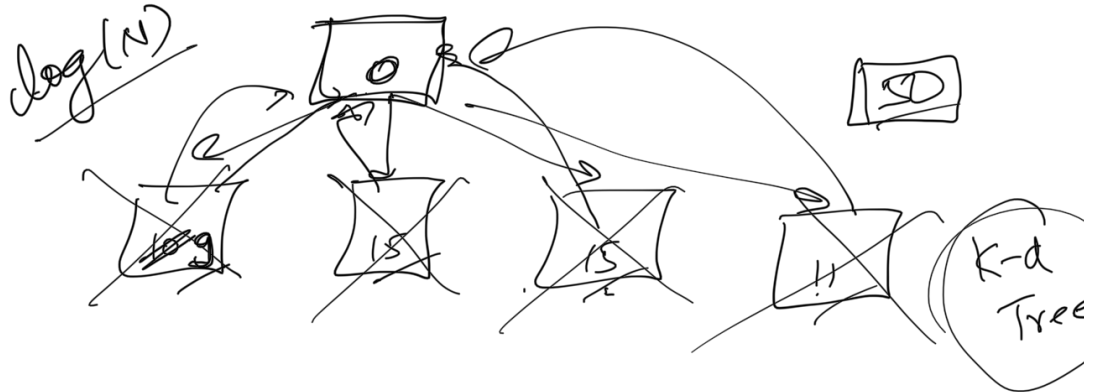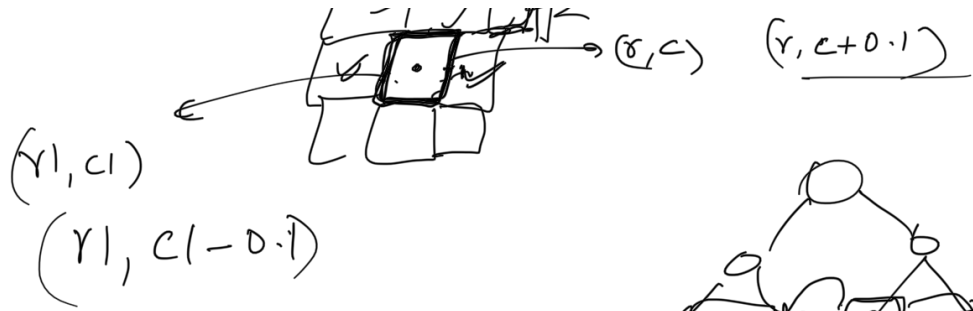find-leaf (root, coord)
  if root is leaf :
    return root

  c1 is the right
  return find-leaf (c1, co...

(r, c) $(r, c+0.1)$

$(r1, c1)$

$(r1, c1-0.1)$

$\log(N)$

0

$\log$ $\mid S \mid$ $\mid S \mid$ $!!$

K-d Tree

if parent has 4 leaf co
Children

and $c1 \cdot places \cdot size + c2 \cdot p$
$+ c3 \ldots + c4 \ldots - c$

---

restaurants $\longrightarrow$ grid.

root $\rightarrow$ grid

Quad Tree

SQL
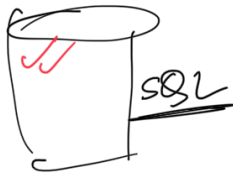
6 billion

g1 ...

1 billion places of interest
↳ restaura
↳ hotel

1 billion

S1   S2   Sk

50 places.

1000

1 billion

$$\overline{5\cancel{8}25}$$ $\overline{\text{40 million leaves}}$

$+10M + \dfrac{10M}{4} + \dfrac{10M}{16}$
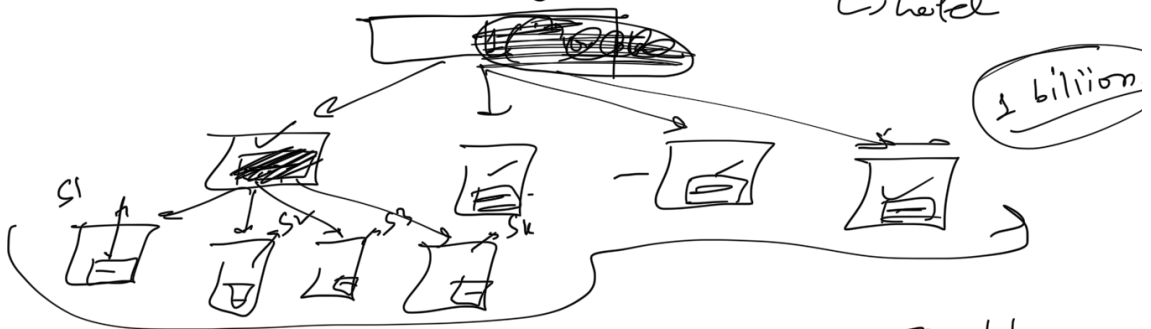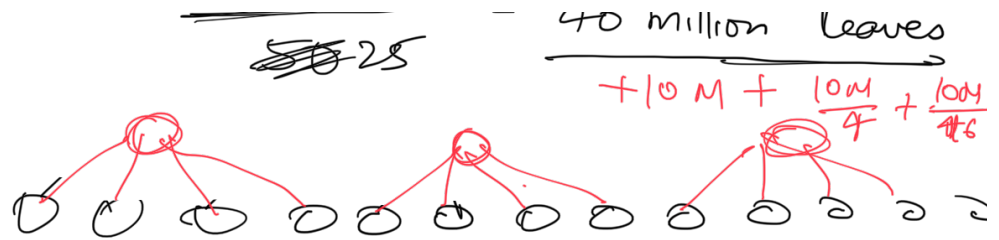
$\underline{53M}$

$$40M * \left(1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \cdots \right)$$

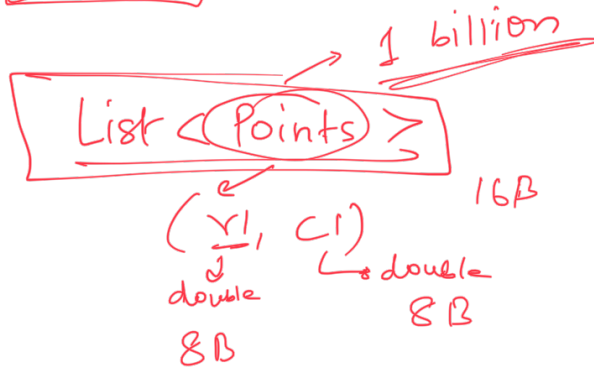$$\lesssim 40M * \left(\frac{1}{1-\frac{1}{4}} \right) = 40M * \frac{4}{3}$$

$$\approx \cancel{50} \boxed{50} M$$

$$\underline{53M} * \underline{16\,bytes} + \boxed{1\,billion} * 16\,bytes$$

$$\approx \boxed{17\ GB}$$

$8\,bytes$

$(lat,\ long)$
$\downarrow$
$8\,bytes$

$\boxed{List\ \langle Points \rangle}$ $\nearrow$ 1 billion

$16B$

$\left(\underline{r1},\ c1 \right)$
$\downarrow$
double
$8B$
$\hookrightarrow$ double
$8B$

Quad
Tree
$\downarrow \cancel{\times}$
Shardin