3,
;

3.
then, Chatgpt somehow mentioned, that LinkedHashMap
is not thread Safe

Then I recollected, oh yes, Since hashMap is not thread safe
hence LinkedHashMap is also not thread Safe,

I asked for an example, where its not thread safe

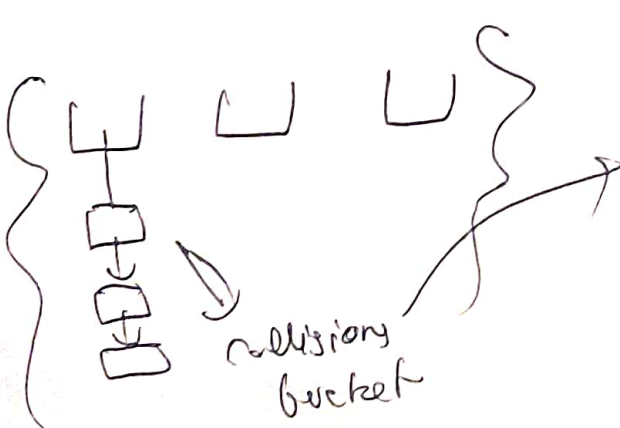So for that lets look at the loadFactor parameter of Hashmap.

.... Hashmap (intt initialCap, ~~float~~ float loadfactor)

→ Well this was initial Cap = 16 → Lets say,

load Factor = 0.75 f = 3/4.

So as the Hashmap fills upto $\left(0.75 f\right) \times 16 = 12$.

the Hashmap will resize.

We know Hashmap is like array of buckets

collision
happes then linkedlist
is formed.

Now when, resizing is happening, the HashMap finds an
array of contigous segment of double its size,
During resizing, copying of contents, lets say that

collision
bucket

Now when getting copied to new
&& location

lets say two threads
T_1, T_2 were doing put operation
and both triggered resize.

P.T.O

(If thread1 || thread2) $\longrightarrow$ whichever thread is

doing put operation,

Bucket1   node1 → node2 ↮ null   is responsible for the

Bucket2   node1   resize.

Bucket3

$$(node1.next = node2 \text{ null}) \rightarrow thread1,$$

And maybe accidentally

$$(node2.next = node1)$$
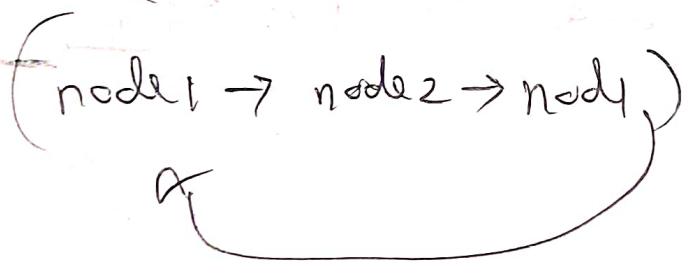
$$(node1 \rightarrow node2 \rightarrow node1)$$ formation of all infinite loop, hence thread safety is important