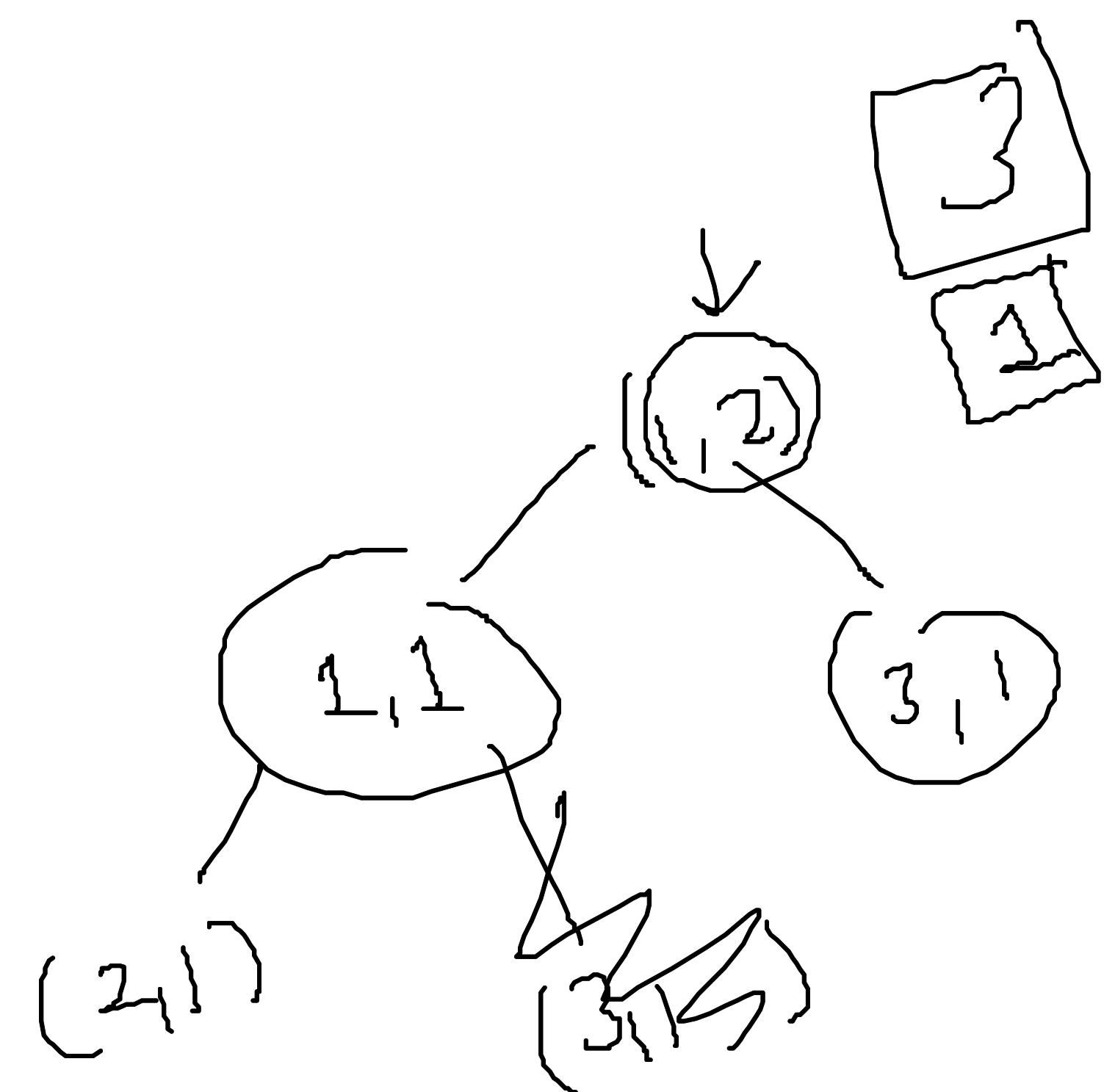


1, 2, 3, 3, 4

1, 2, 3, 1, 3, 2, 3, 4



Top
k=2

freq
1 - 2 ✓
2 - 1 ✓
3 - 0 ✓
4 - 0

Hashing & Tries

↓
hashtable

Insert(K,V)
Search(K) ← $O(1)$ on average
Delete(K)

Key-value pairs
Dosa - 30
Massi - 20
shake - 50

fast
Unordered

Implementation

int, Value

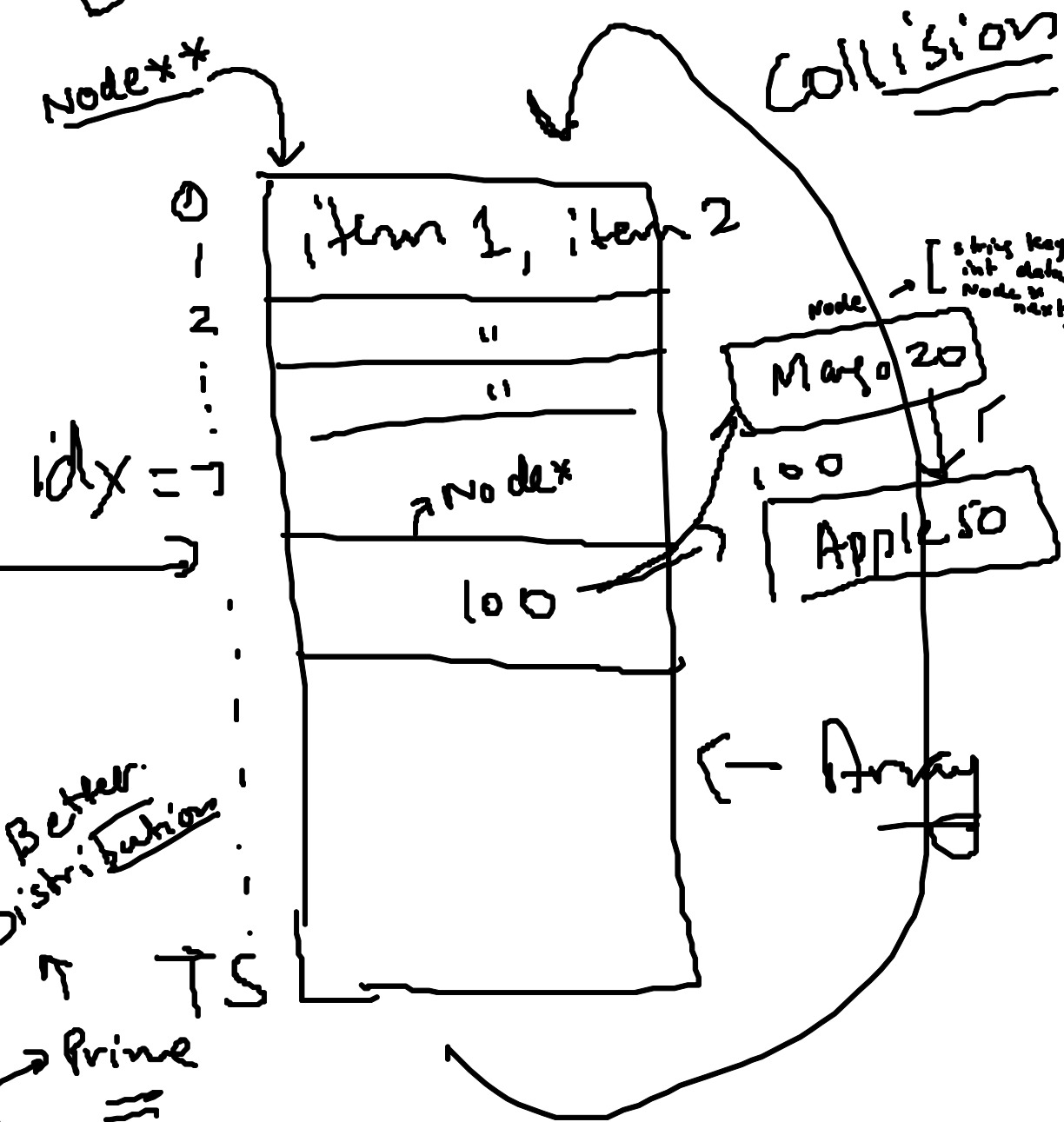
2[5] = True

Key
✓ String

Key = String
value = Any type of Data/object

Apple: 50
String = Apple

hashFn



avoid x
Reduce the chance of collision by using a good hash fn.

idx → TS
→ % TS
O(1) time
TS = 11
idx = 22, 33

Load Factor of Table

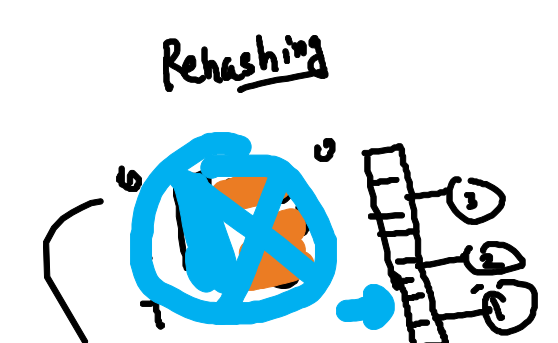
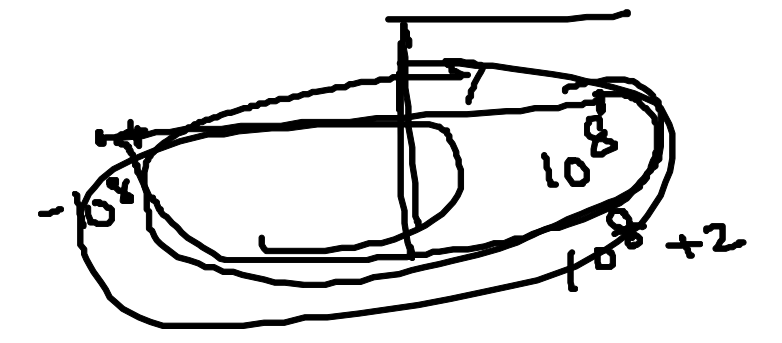
ABCD → (65 + 66 + 67 + 68) % TS
BACD → collision.

bat → tab

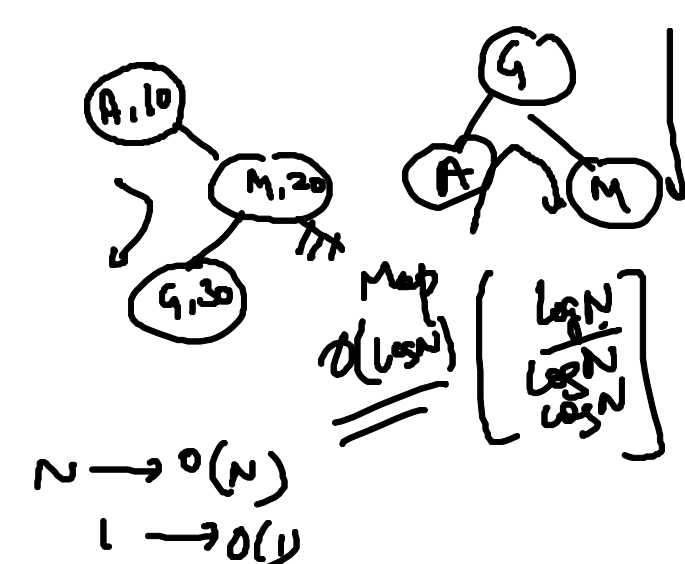
65, 66, 67, 68
(65 · 10³ + 66 · 10² + 67 · 10¹ + 68 · 10⁰) % TS
10, 20, 30, 100, 200, 1
Collide
overflow

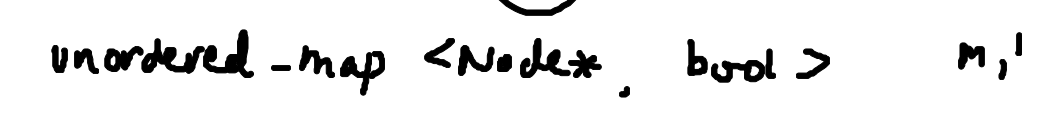
(a × b) % m

= (a % m × b % m) % m



Load Factor = $\frac{\text{Current Size}}{\text{Total Buckets}}$
= $\frac{6}{7} = 0.85$





777777
 $0, 2, 5, 4, 3, 6$ → sorted
 $0, 1, 3, 10$ → Union
 $map < int, bool >$
 $h[0] = true$
key

