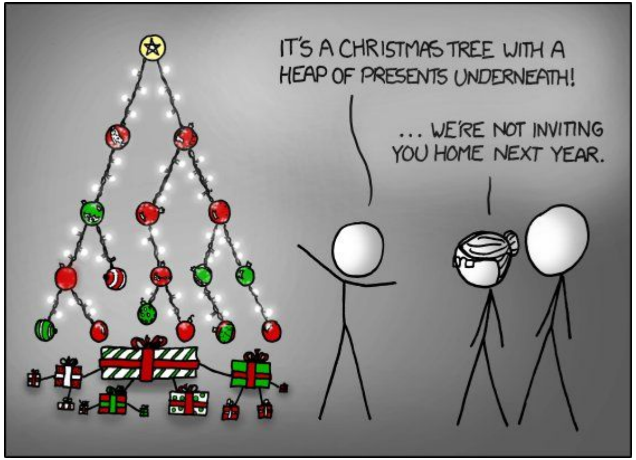
Learning-Augmented Skiplists

Presenter: Jung, Chunkai
Project supervised by Dr. Samson Zhou



***Disclaimer**: the contents herein are used solely for academic purposes. If you find any of the contents in these slides involves copyright issues, please contact the authors first, we will resolve it asap.*

Contents

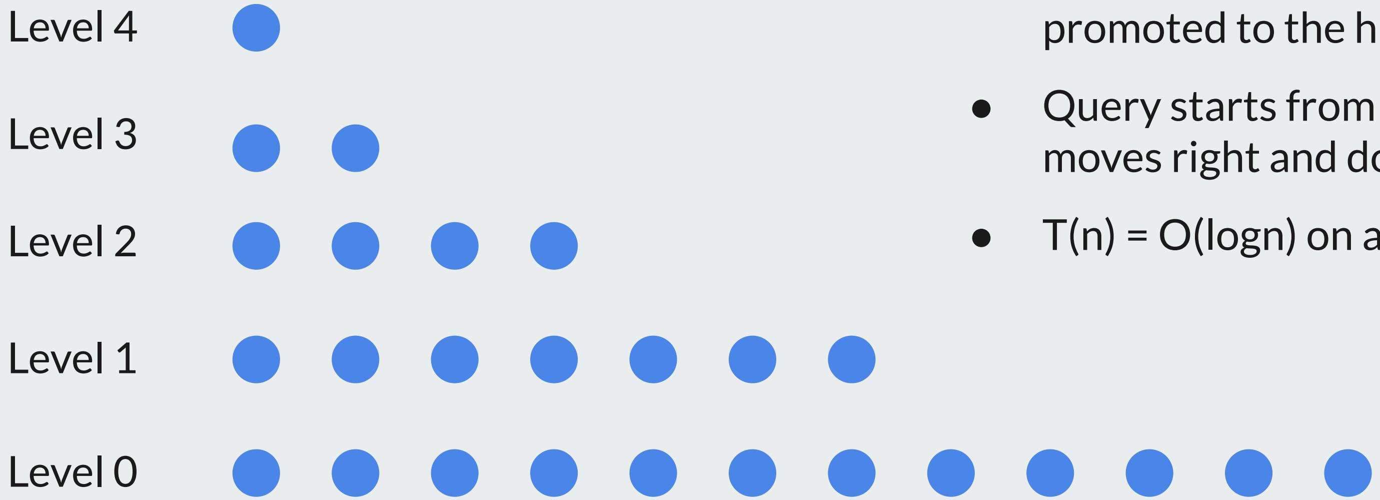
Why skip lists when you could use a tree?

- Skip lists are easier to implement and maintain with similar performance as BST.
- Skip lists are more efficient on range queries and frequent queries than BST.
- Skip lists allows for easier implementation of lock-free and fine-grained locking mechanisms.
- Skip lists typically use less space than balanced binary trees.
- Skip lists have better cache performance than trees (linked lists and memory locality).
- For distributed systems, random promotion are preferred over deterministic balancing.
- For multi-threaded applications, skip lists are more amenable to concurrent access.
- … …

Problem Formulation

Given a set of elements and the probability of each element appearing in the query stream, build a learning-augmented skip list data structure so as to improve querying operations on the structure.

Vanila Skiplist

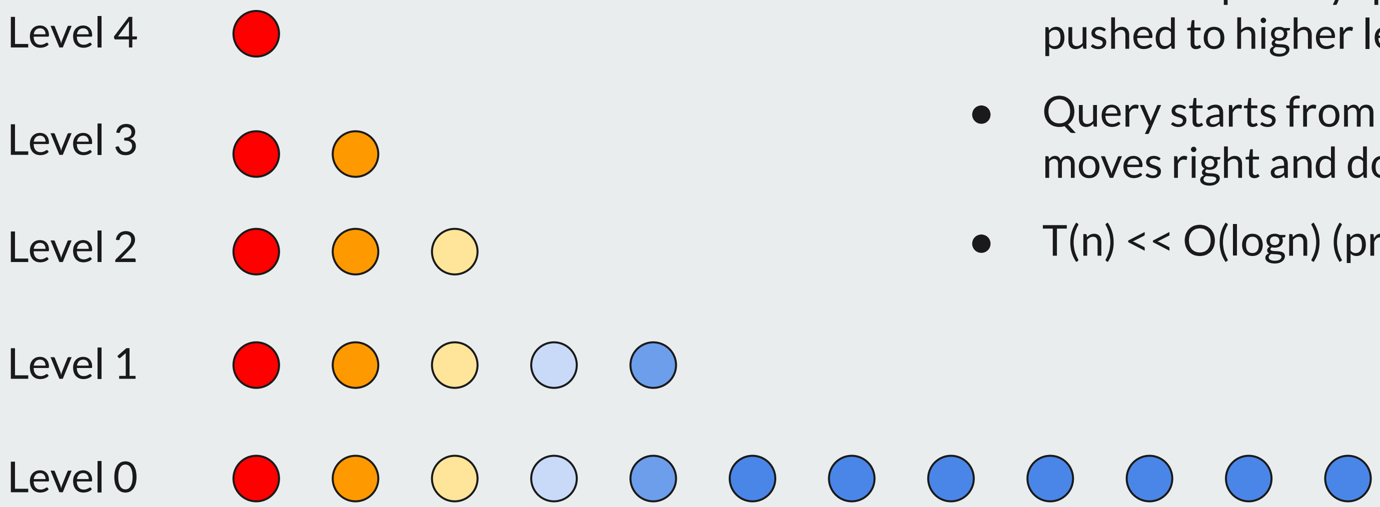


Facts:

- Every node has a 50% chance of being promoted to the higher level.
- Query starts from the highest level and moves right and down similar to BST.
- T(n) = O(logn) on average.

All elements are assumed to be equally important, the data structure has no extra info.

Learning Augmented Skiplist



Facts:

- More frequently queried node will be pushed to higher levels.
- Query starts from the highest level and moves right and down similar to BST.
- T(n) << O(logn) (proved theoretically).

Elements are not equal, each element has additional info: probability of appearance in query.

Learning Augmented Node Promotion Algorithm

1.1 Algorithm

Algorithm 1 Learning-augmented skip list

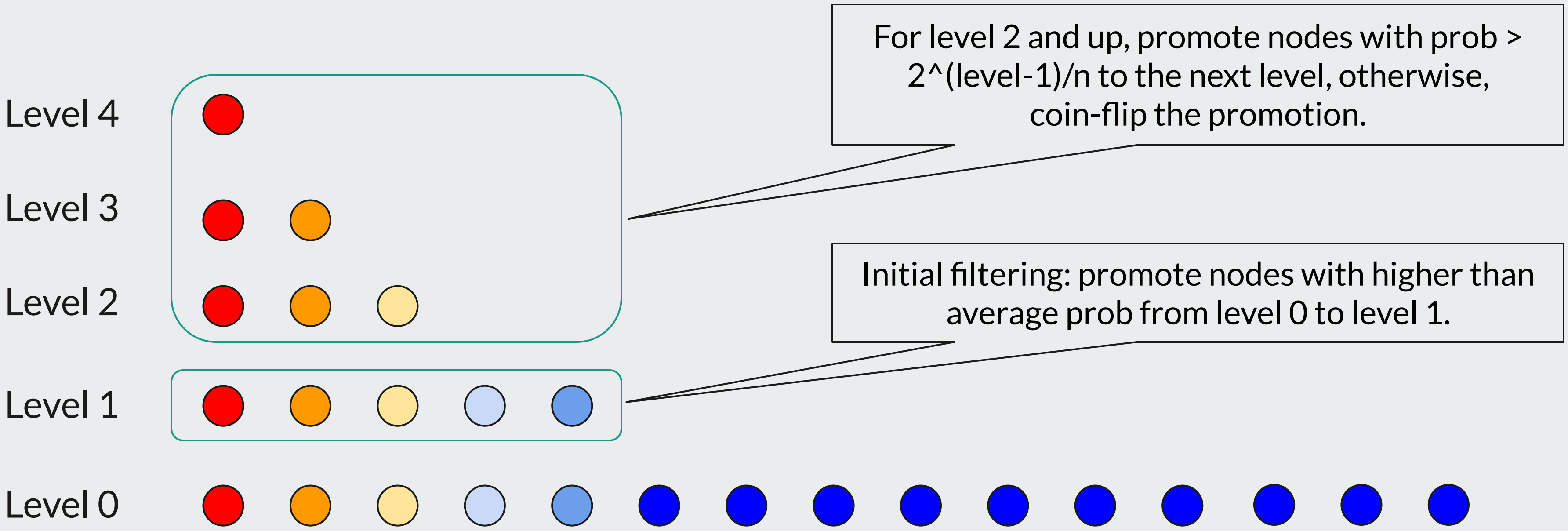
Input: Predicted frequencies $p_1, \ldots, p_n$ for each item in $[n]$
Output: Learning-augmented skip list

```
Insert all items at level 0
Insert exactly the items with predicted frequency at least 1/n at level 1
for each ℓ ∈ [2 + log n] do
    for each i ∈ [n] do
        if predicted frequency p_i ≥ 2^(ℓ−1)/n then
            Insert i into level ℓ
        else if i is in level ℓ − 1 then
            Insert i into level ℓ with probability 1/2
```

ID, Value

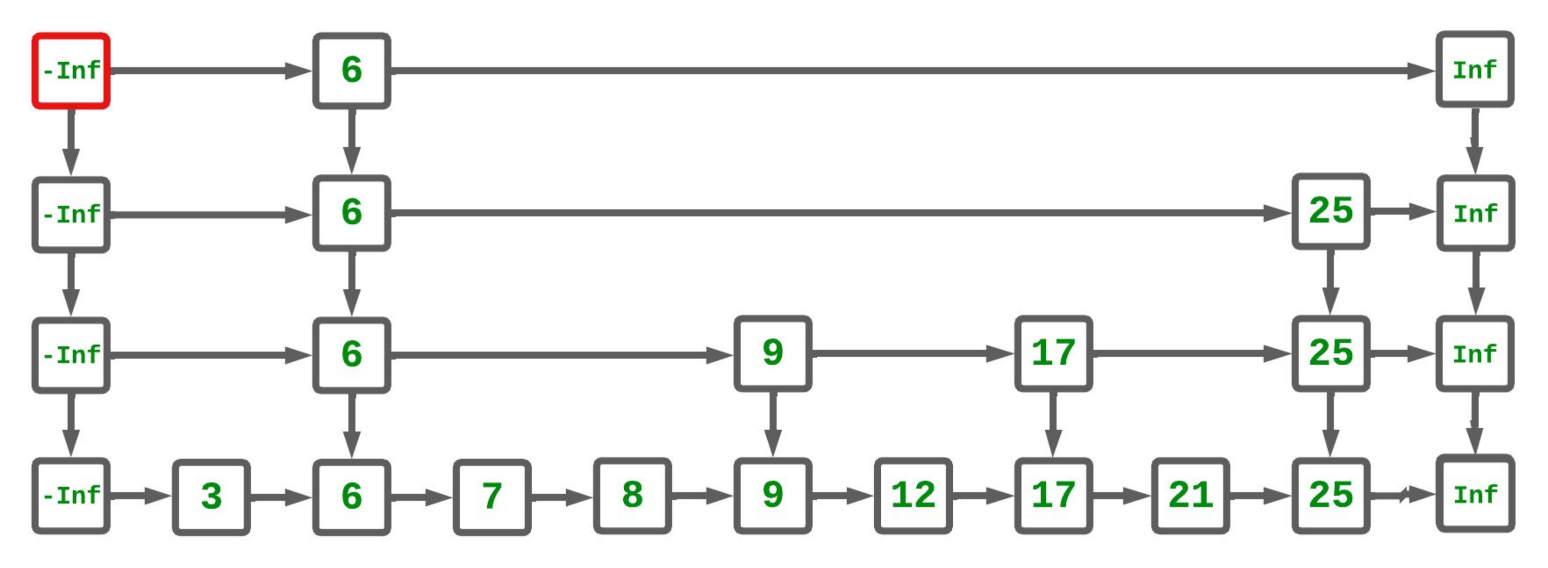
ID, Value, Prob ...

Learning Augmented Node Promotion Demo



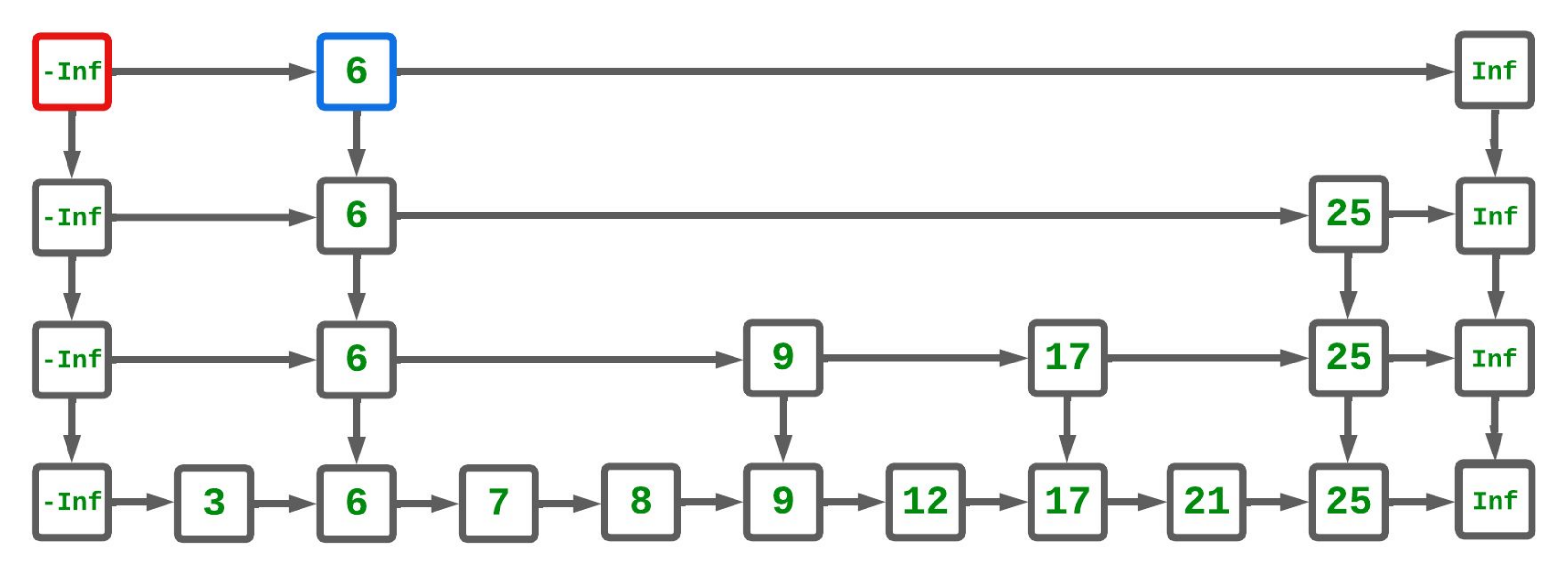
Skiplist Operation - Insertion

Inserting a value 26



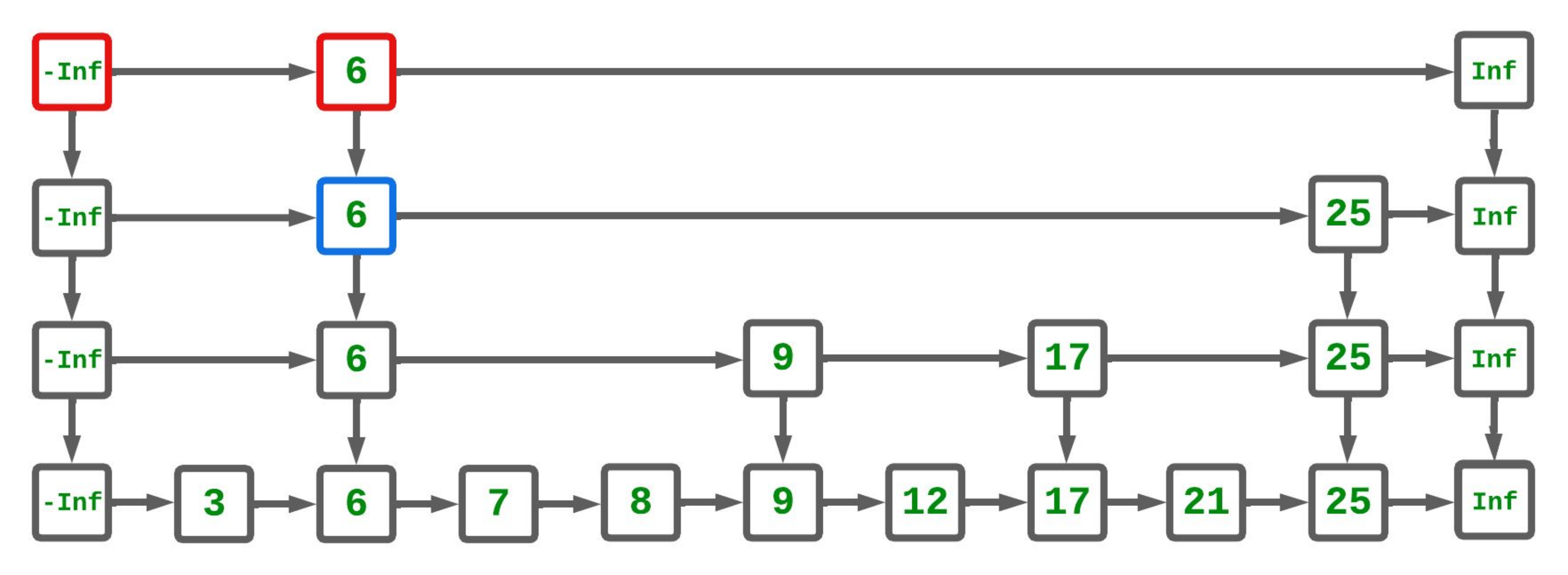
Skiplist Operation - Insertion

Inserting a value 26



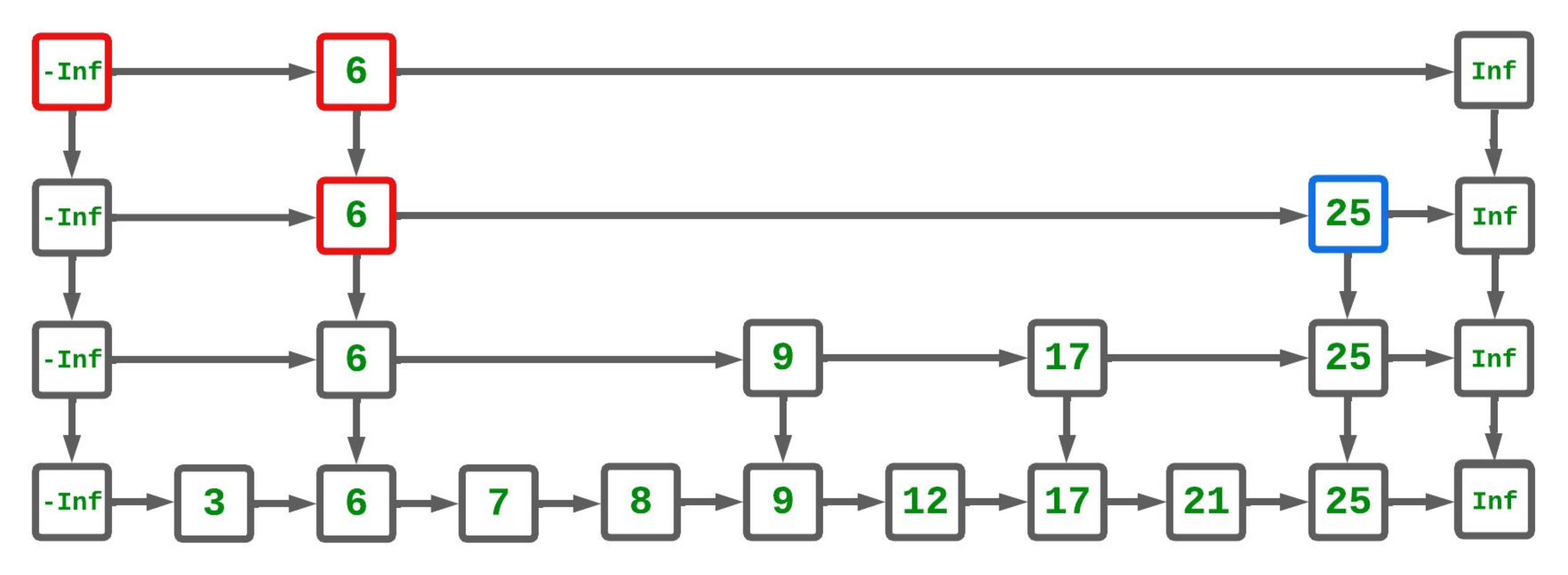
Skiplist Operation - Insertion

Inserting a value 26



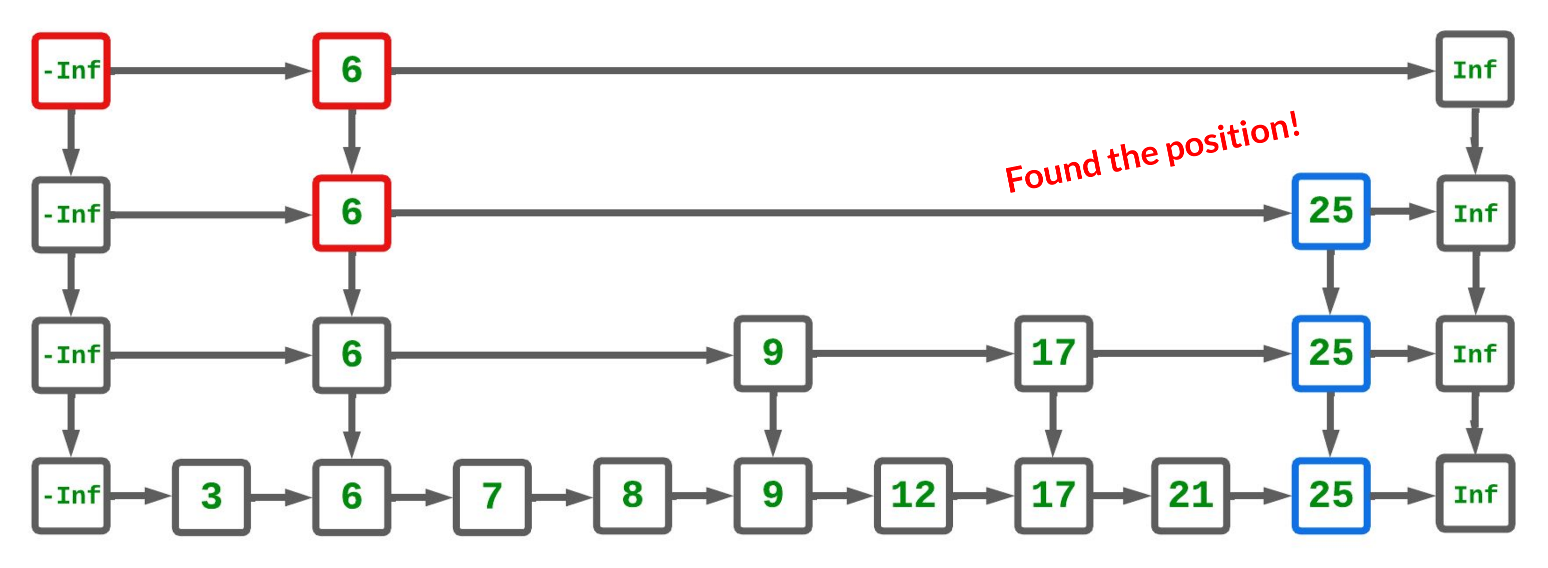
Skiplist Operation - Insertion

Inserting a value 26



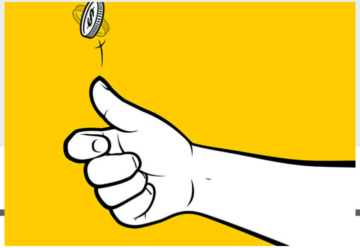
Skiplist Operation - Insertion

Inserting a value 26



Skiplist Operation - Insertion

Inserting a value 26



Head => Add Nodes
Tail => Stop

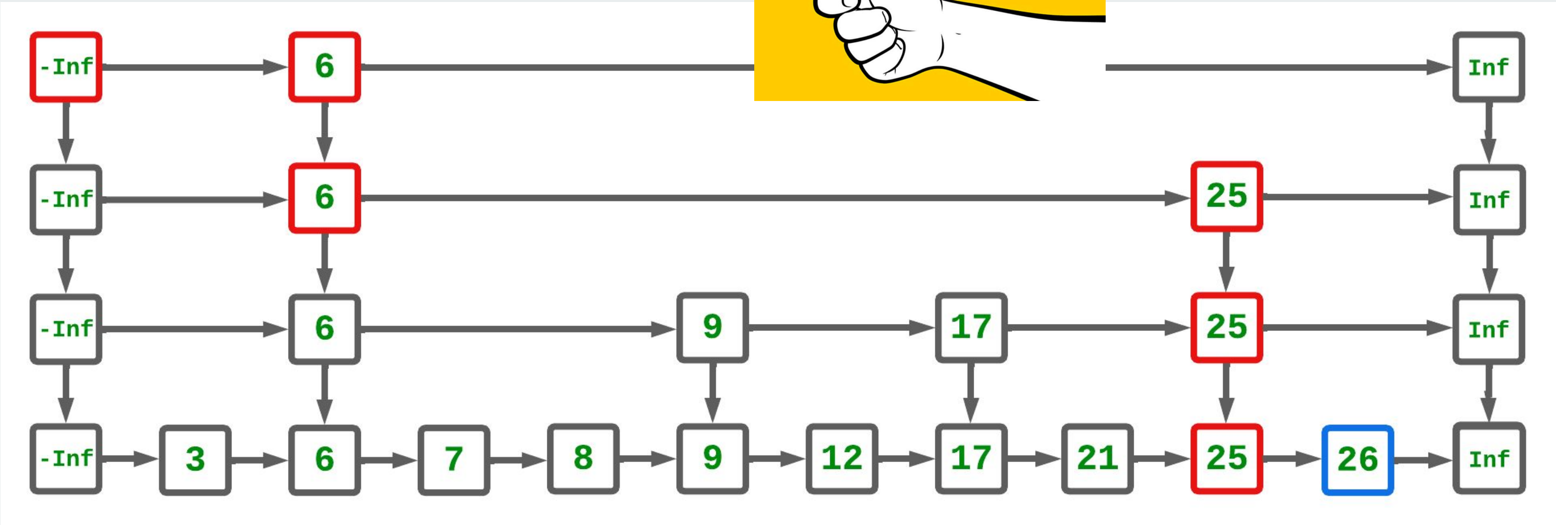


Image: https://nationaltoday.com/flip-coin-day/

Skiplist Operation - Insertion

Inserting a value 26

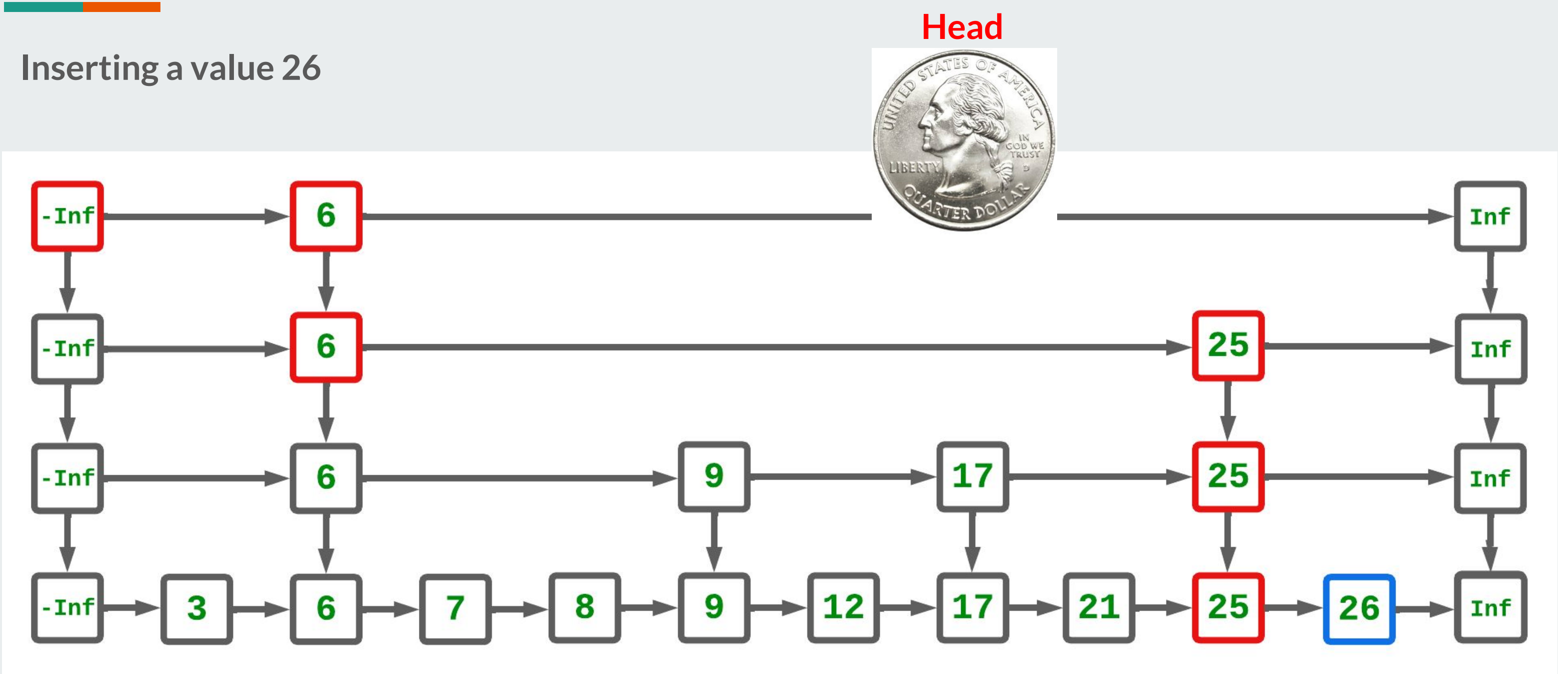
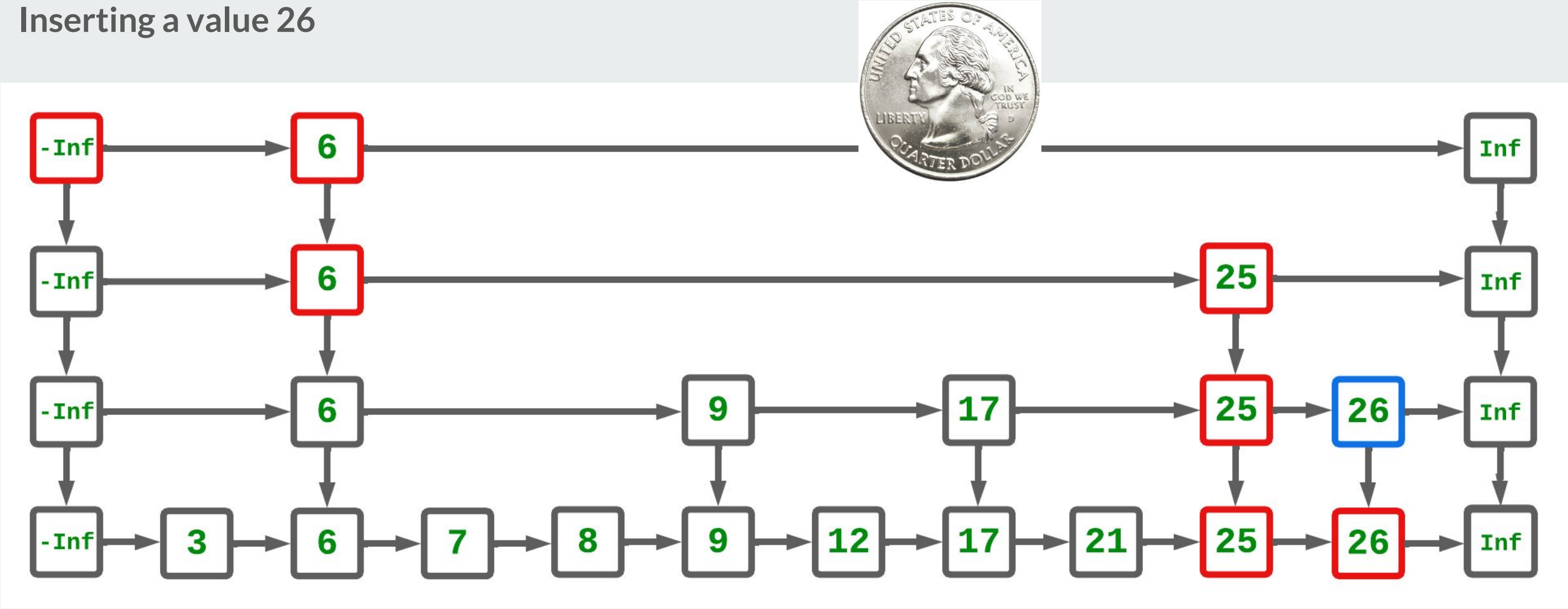


Image: https://nationaltoday.com/flip-coin-day/

Skiplist Operation - Insertion

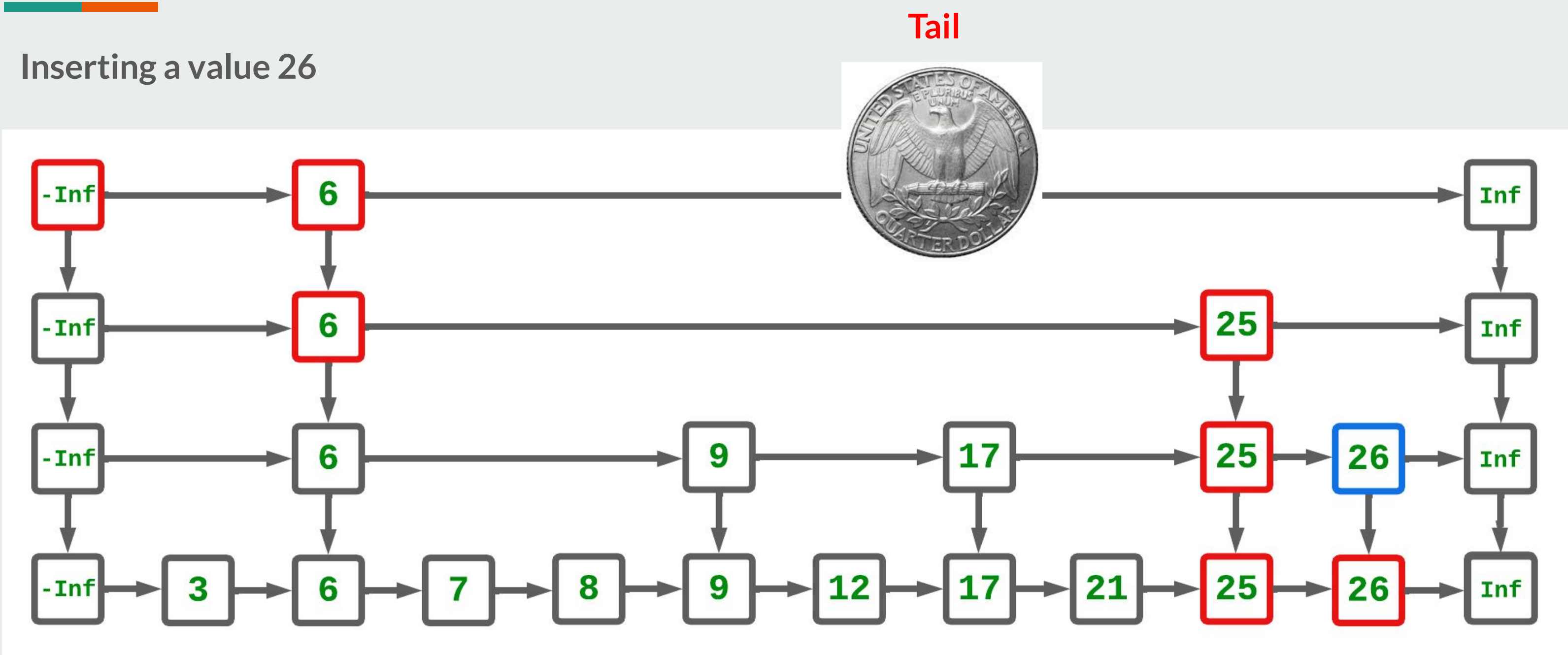
Inserting a value 26

Head



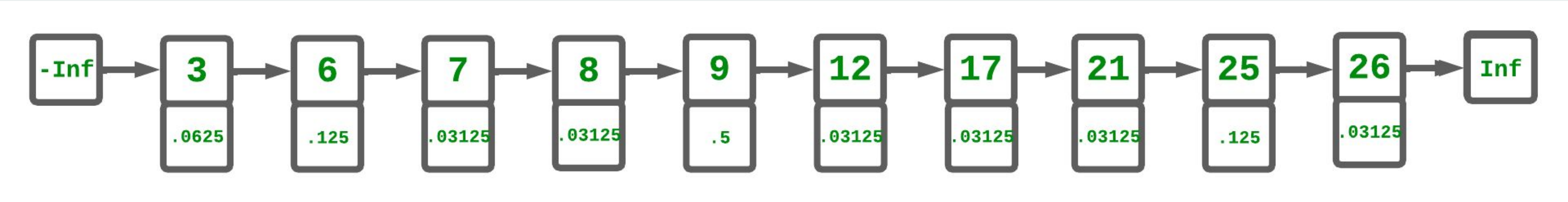
Skiplist Operation - Insertion

Inserting a value 26



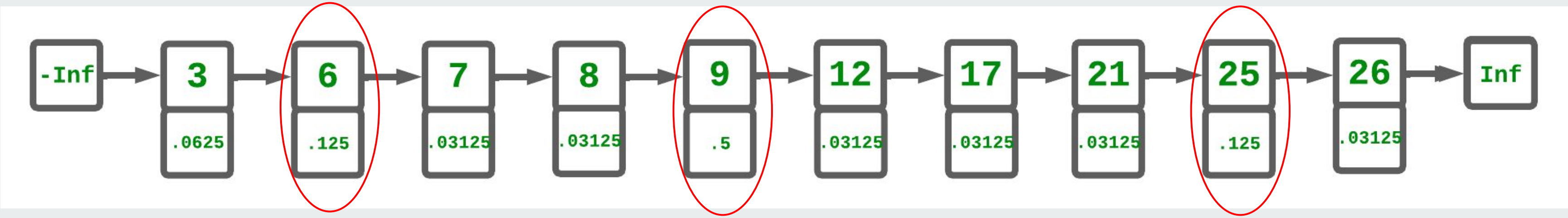
Learned Skiplist - Insertion

Level 0: Insert all elements at level 0



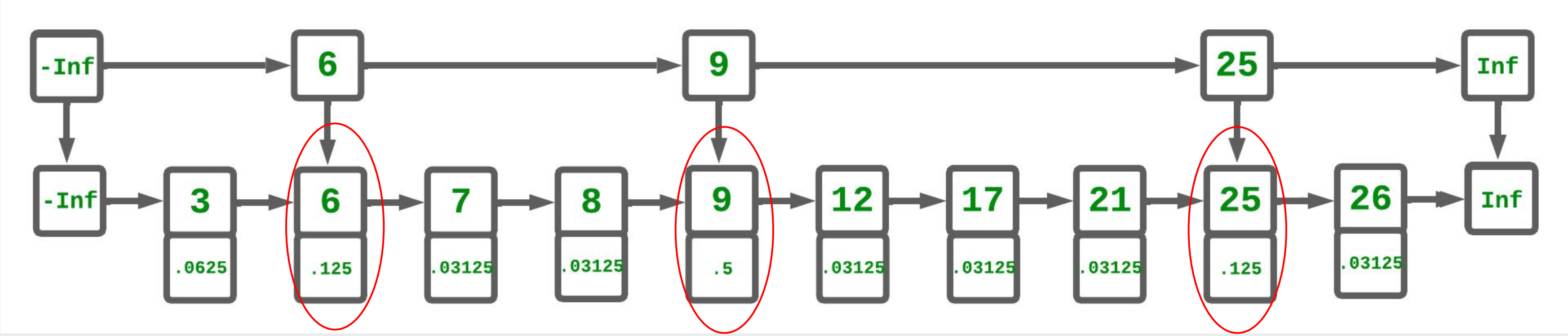
Learned Skiplist - Insertion

Level 1: Insert the items with predicted frequency at least 1 / n (i.e., 0.1)



Learned Skiplist - Insertion

Level 1: Insert the items with predicted frequency at least 1 / n (i.e., 0.1)

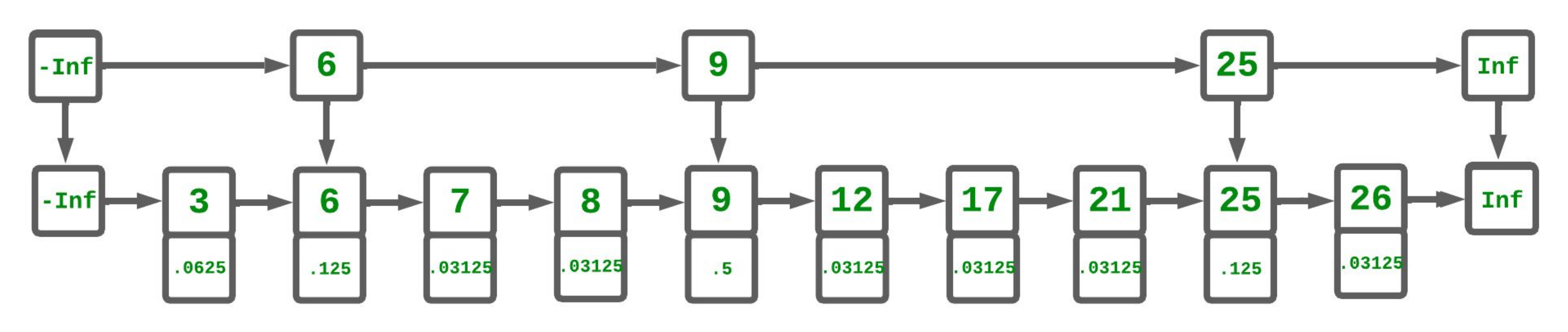


Learned Skiplist - Insertion

Level 2: Insert the items

If : predicted frequency $p_i \geq \frac{2^{\ell-1}}{n}$

Else: Coin flip

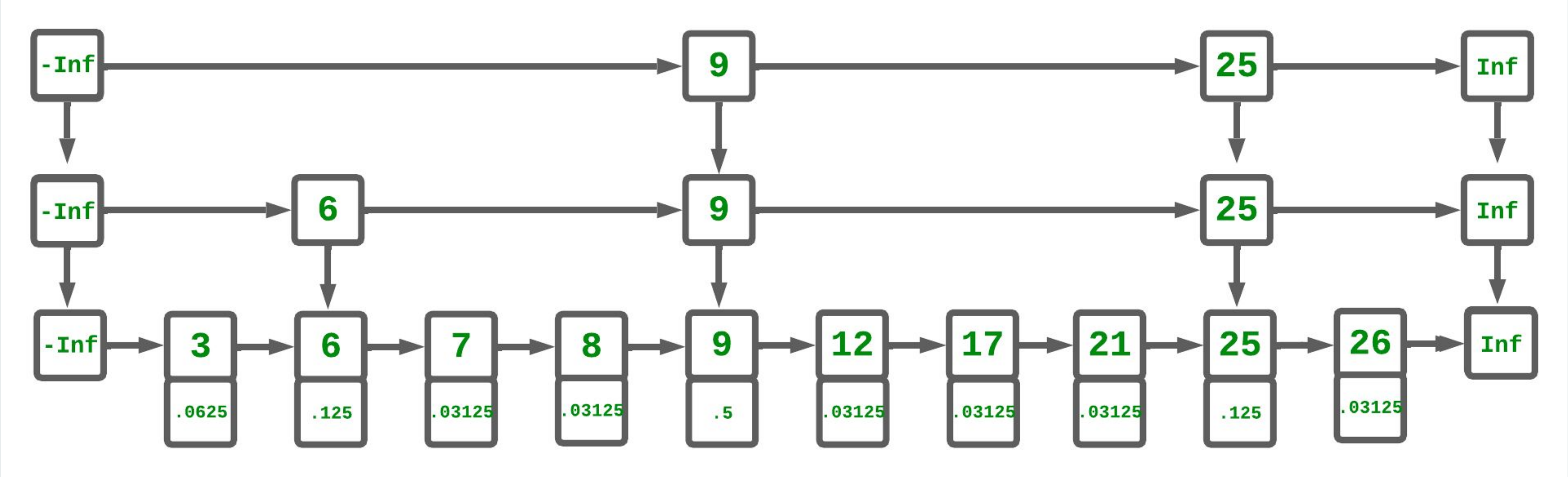


Learned Skiplist - Insertion

Level 2: Insert the items

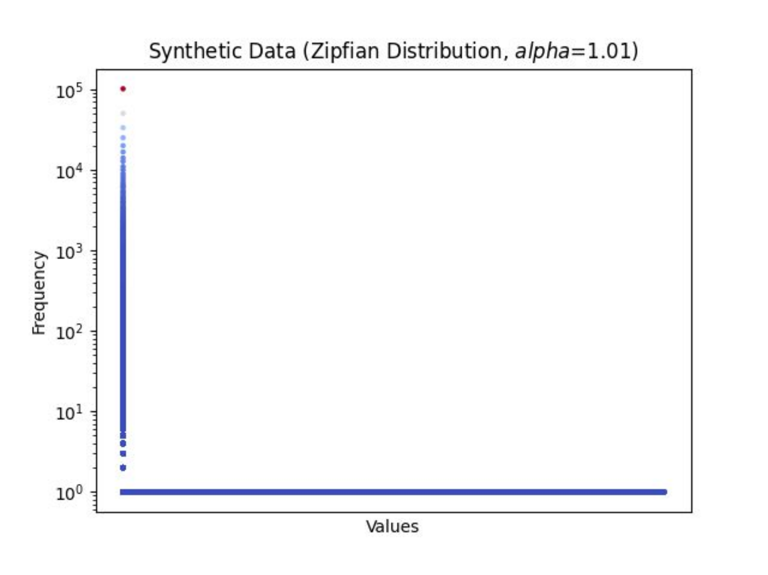
If : predicted frequency $p_i \geq \frac{2^{\ell-1}}{n}$

Else: Coin flip

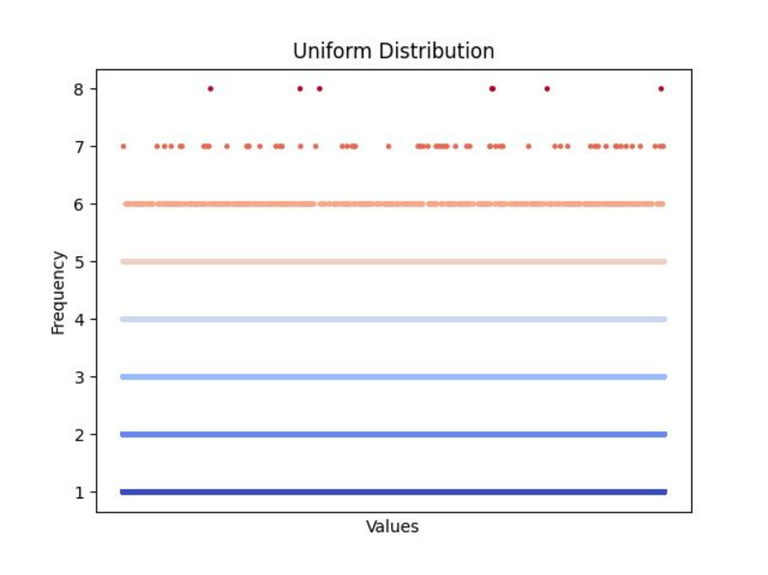


Results on Synthetic Datasets

Skewed Datasets (represented by Zipfian)

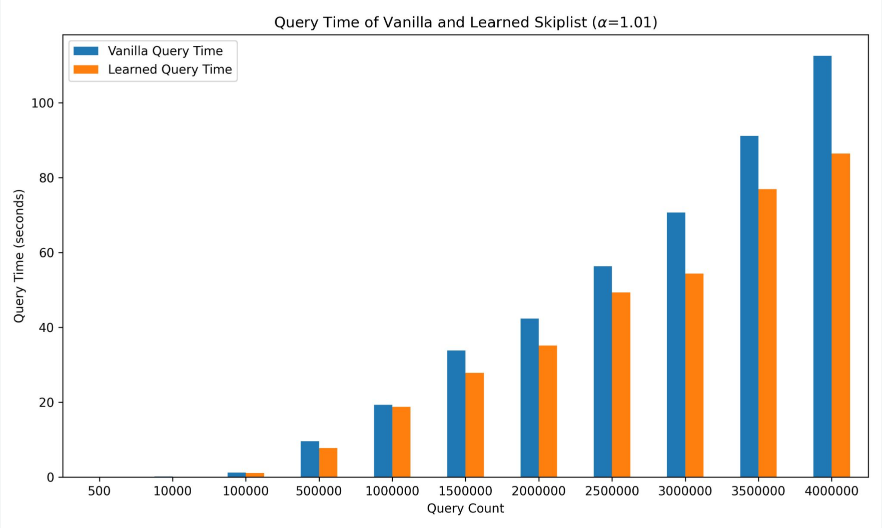


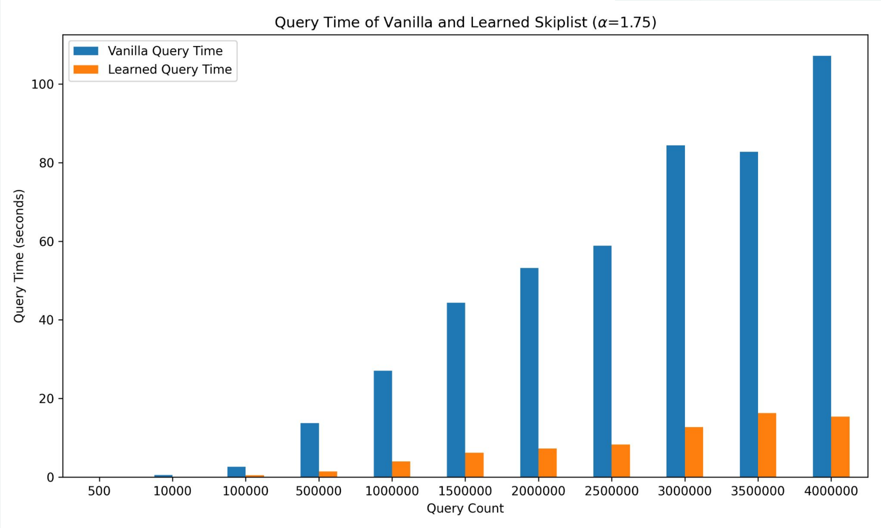
Uniform Datasets



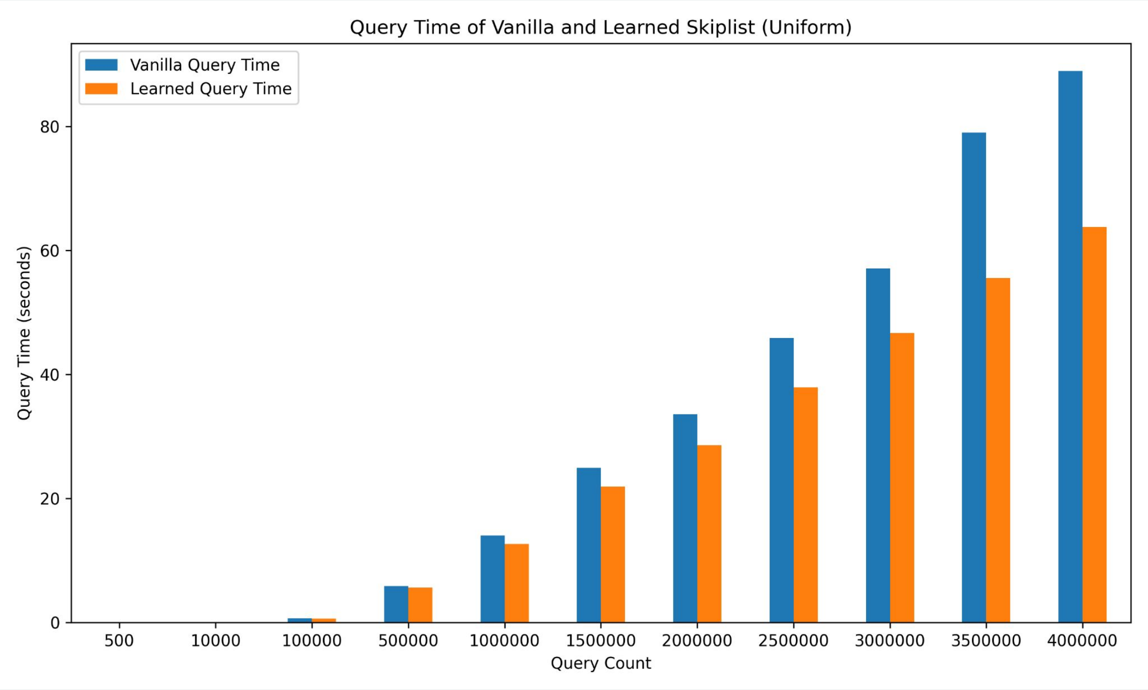
Results on Synthetic Datasets

Alpha	Unique Values
1.01	2886467
1.25	259892
1.5	35539
1.75	8386
2.0	2796





Results on Synthetic Datasets

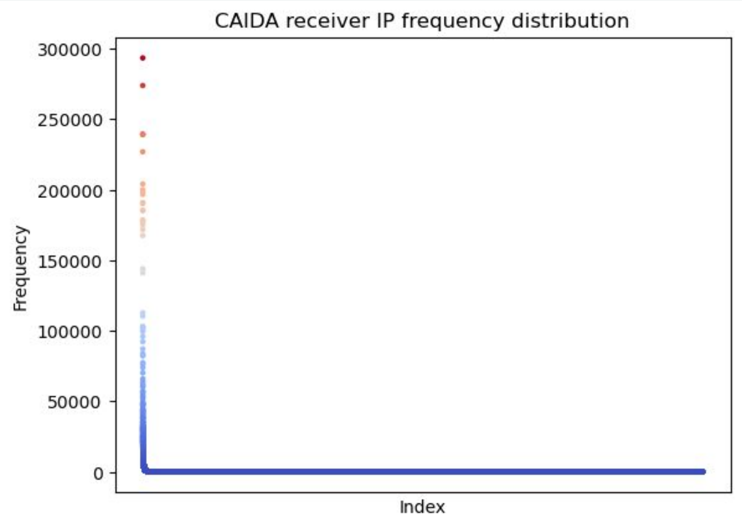


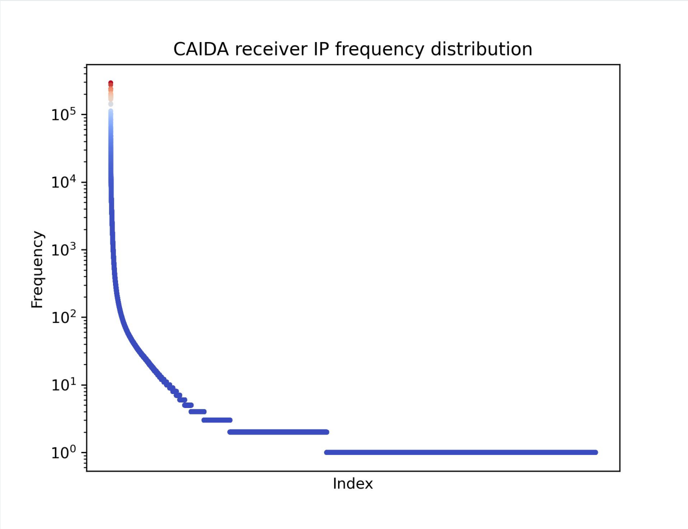
Results on Synthetic Datasets

Speed-up factor of learned skiplist over vanilla skiplist

α	Number of nodes											
	500	10000	100000	500000	1000000	1500000	2000000	2500000	3000000	3500000	4000000	Average
uniform	3.02	0.84	1.01	1.05	1.11	1.14	1.17	1.21	1.22	1.42	1.4	1.33
1.01	3.63	2.6	1.04	1.24	1.03	1.21	1.2	1.14	1.3	1.18	1.3	1.53
1.25	3.28	3.74	5.87	2.89	2.47	3.21	2.95	3.34	3.55	3.16	3.12	3.42
1.5	2.42	8.97	6.93	6.54	7.99	5.83	4.65	3.8	4.92	5.34	5.93	5.76
1.75	12.43	10.4	5.76	9.78	6.76	7.13	7.31	7.09	6.63	5.07	6.98	7.76
2	8.19	2.5	5.56	10.1	4.47	3.91	7.26	5.33	9.29	7.65	5.55	6.35

CAIDA Datasets Distribution

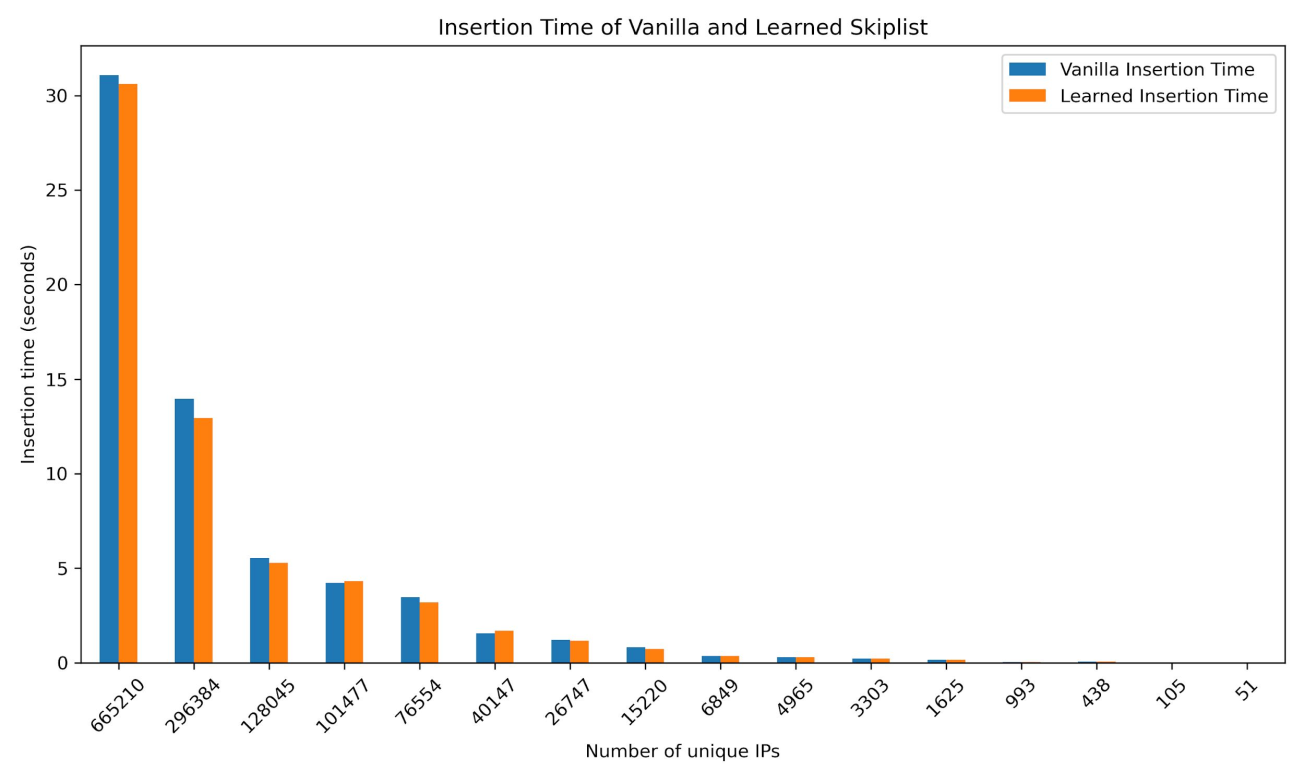




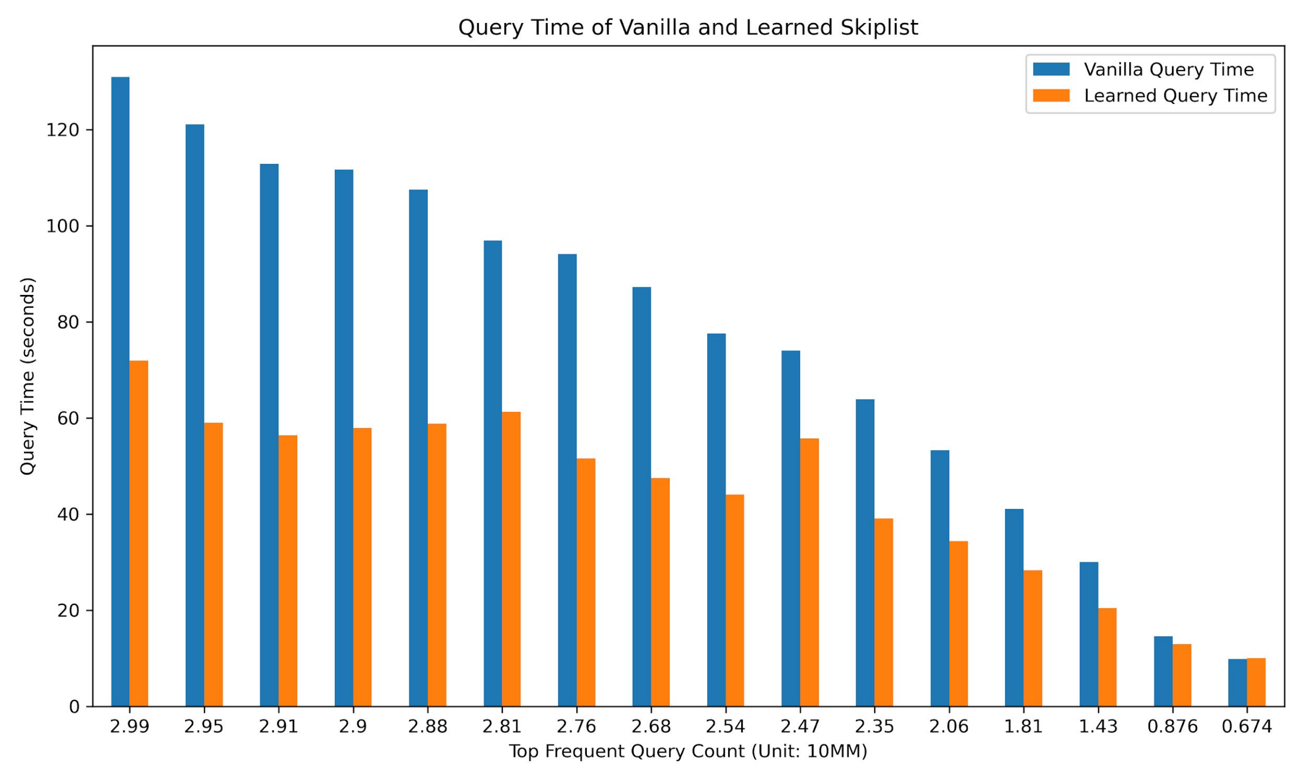
The data shown above contains 29.9 Million IP addresses, with 665,210 unique IP addresses.

Data source: https://data.caida.org/datasets/passive-2019/equinix-nyc/20190117-130000.UTC/

Results on CAIDA Datasets



Results on CAIDA Datasets



Conclusions

- The theoretical proof and experimental results on both synthetic and real world datasets (CAIDA internet trace) show that the learning augmented skip lists outperform a traditional coin-flip skip list in both insertion and query time for either uniform distribution or skewed distributions like a Zipfian distribution.

Questions?
Comments?