Skip-List Evaluation

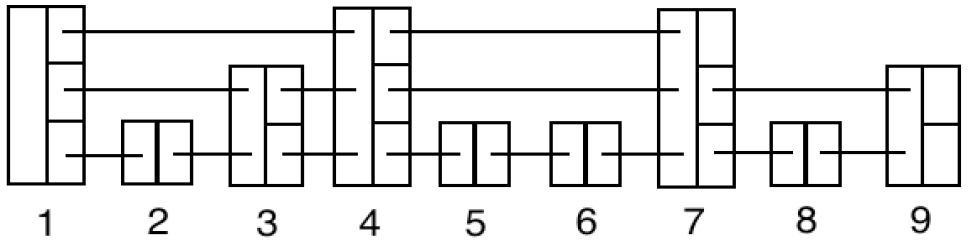
2024.01.17

Presentation by Boseung Kim, Yeongyu Choi

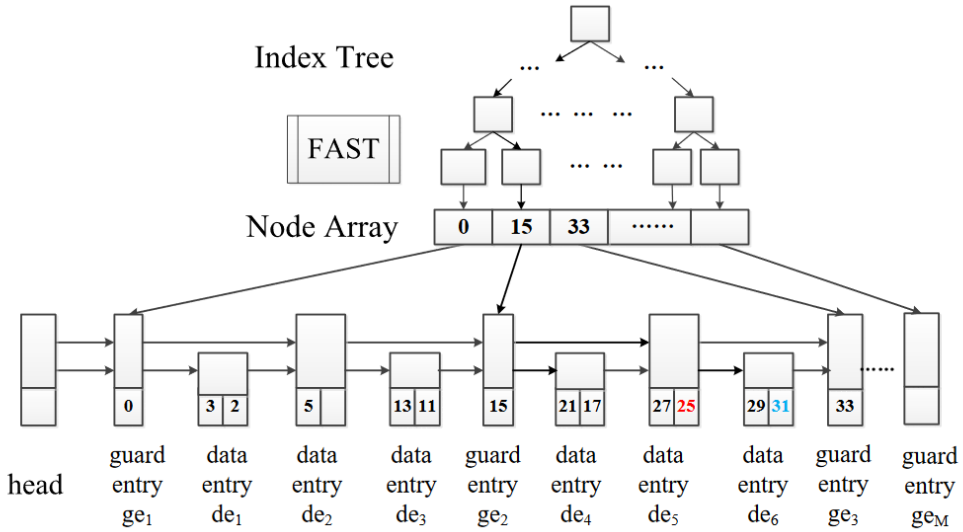
bskim1102@dankook.ac.kr, dusrb1418@naver.com

Contents

Why Skip-List?



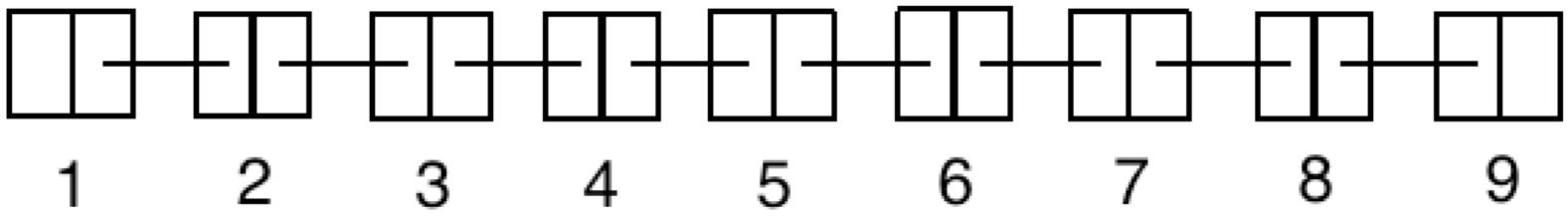
Traditional Skip-List



Semi-order Skip-List

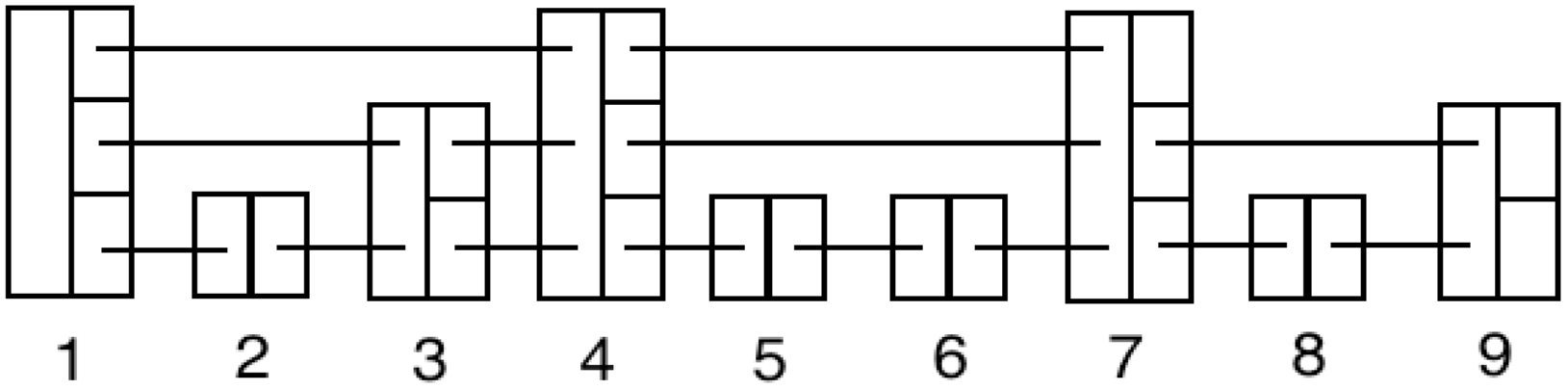
Linked-List

- 데이터 요소를 순서대로 저장하는 선형 자료구조.
  - 배열과 달리 연속적인 공간에 데이터저장을 하지 않음.
  - 각 노드는 독립적으로 메모리에 할당.



Skip-List

- Linked-List를 기반으로 한 데이터 구조.
  - 각 노드는 여러 개의 링크를 가지고 있음.
  - 검색 속도를 높이기 위해 여러 레벨의 링크를 사용함.



Evaluation

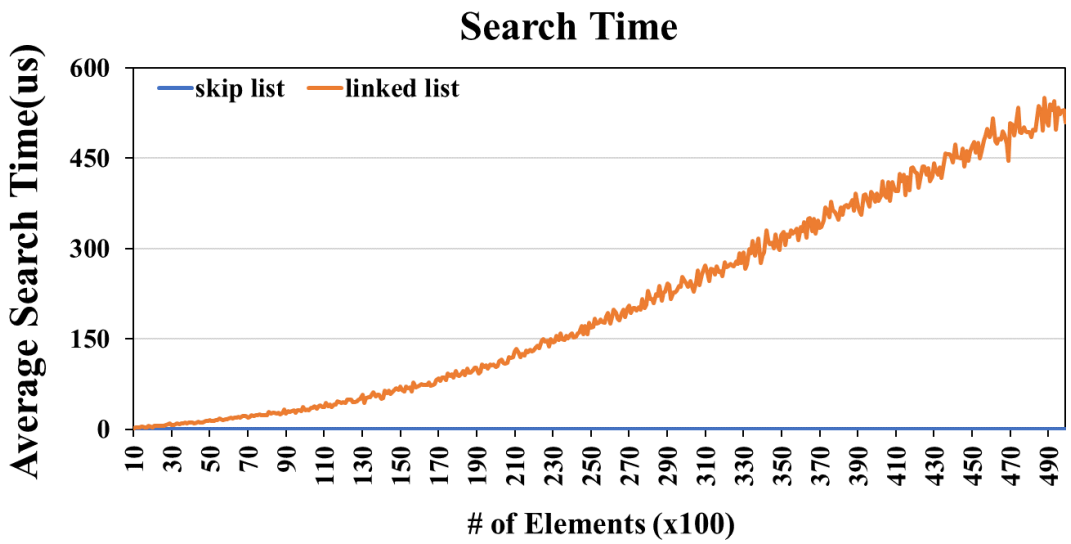
- Skip-List vs Linked-List
- Performance due to Number of Elements
- Insert Method (Random vs Sequential)
- Effect of Max Level
- Effect of Probability

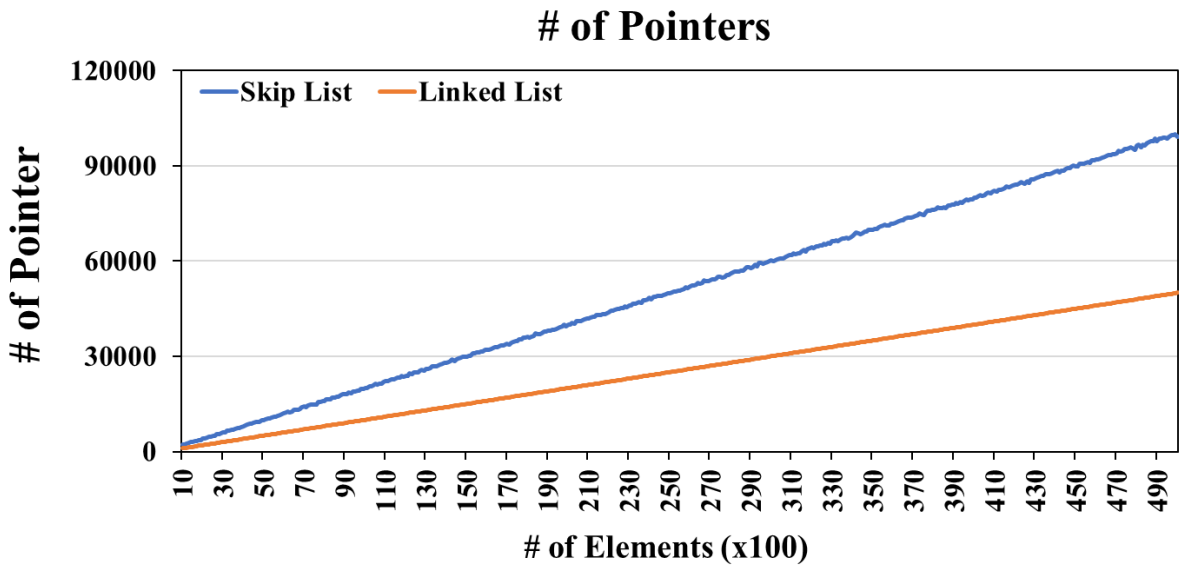
실험 환경

- CPU : 12th Gen Intel(R) Core(TM) i7-12700
  - 25M Cache, 2.10GHz

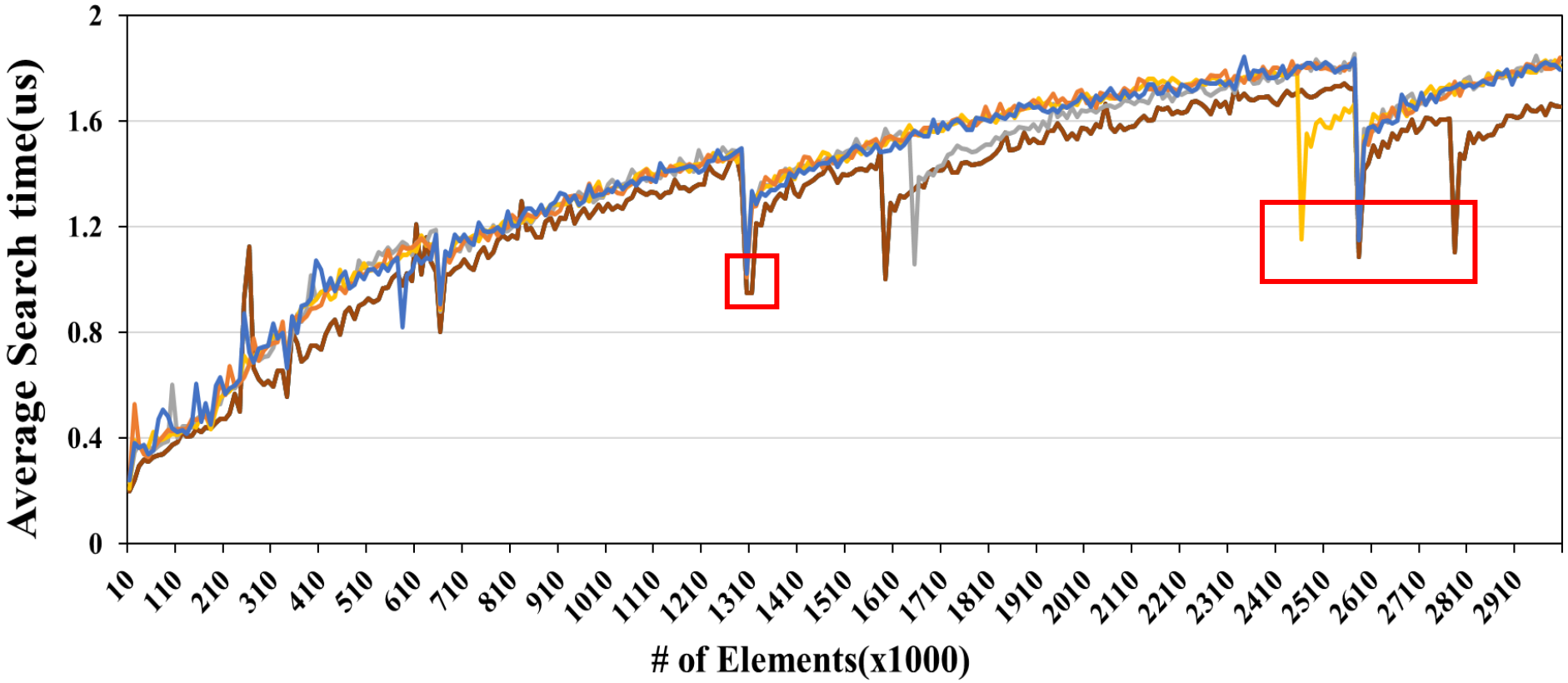
- default:
  - Insert: Sequential, Search: Random
  - Node size: 32byte
  - Probability(p): 50%(0.5)
  - Max_level =$\log_{1/p} n$ $(n = number\ of\ elements)$

Skip-List vs Linked-List

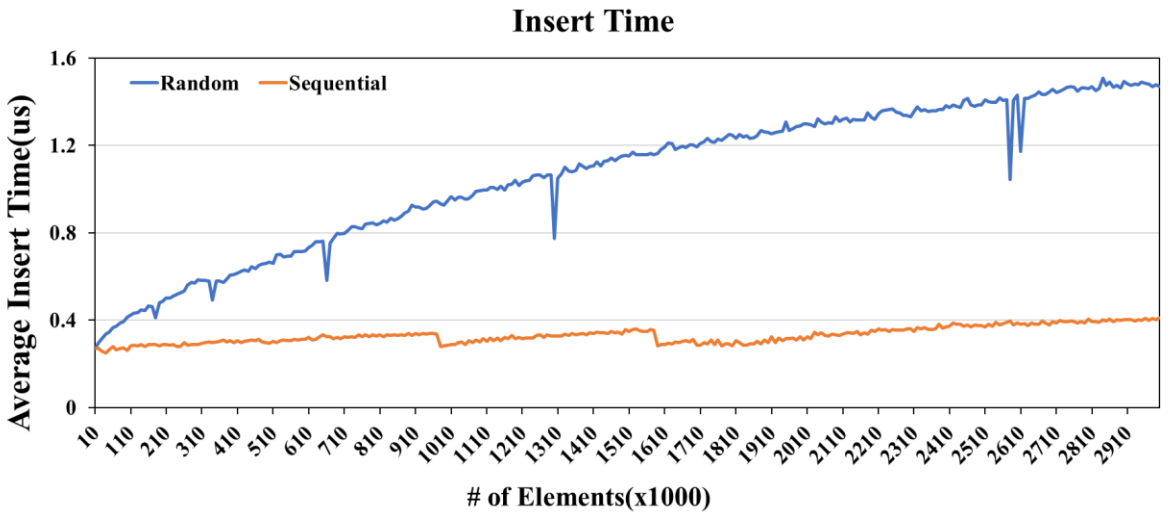


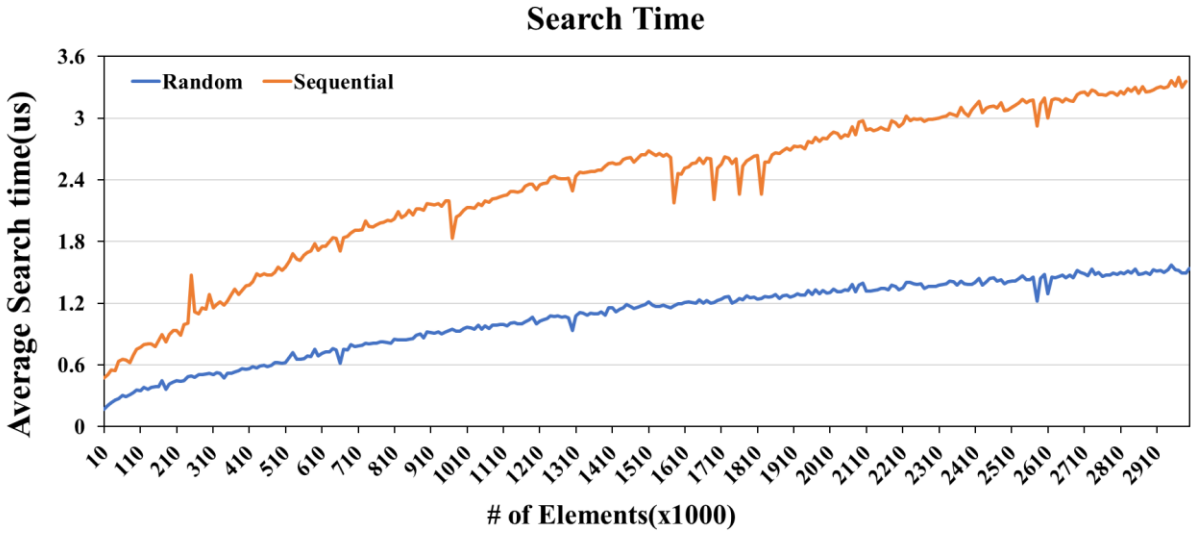


Performance due to # of Elements



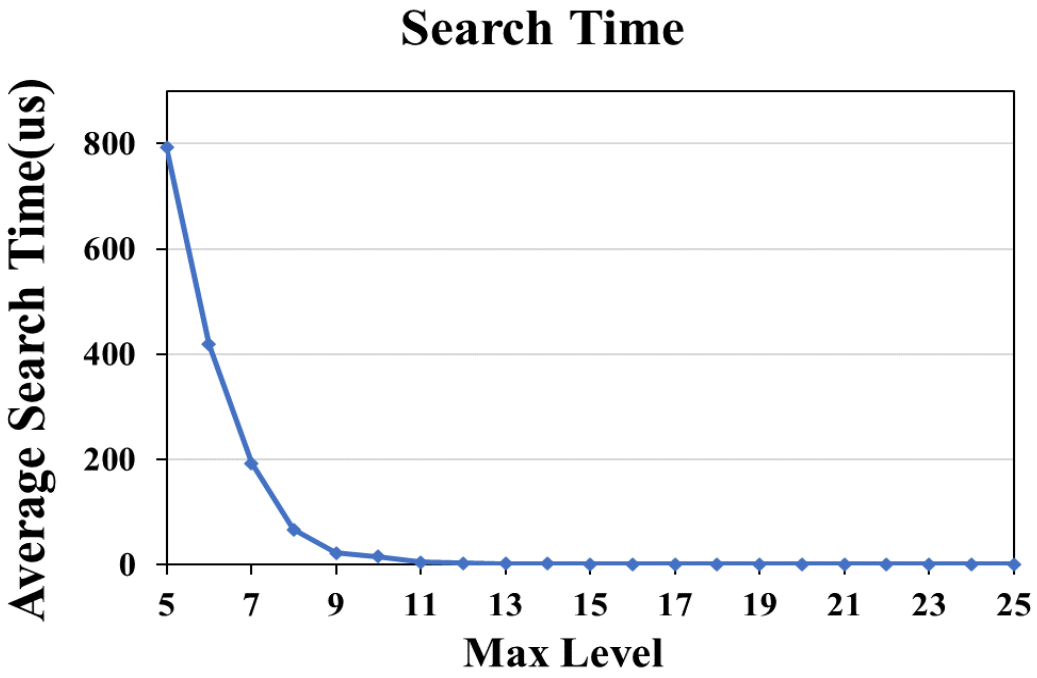
Random Insert vs Sequential Insert

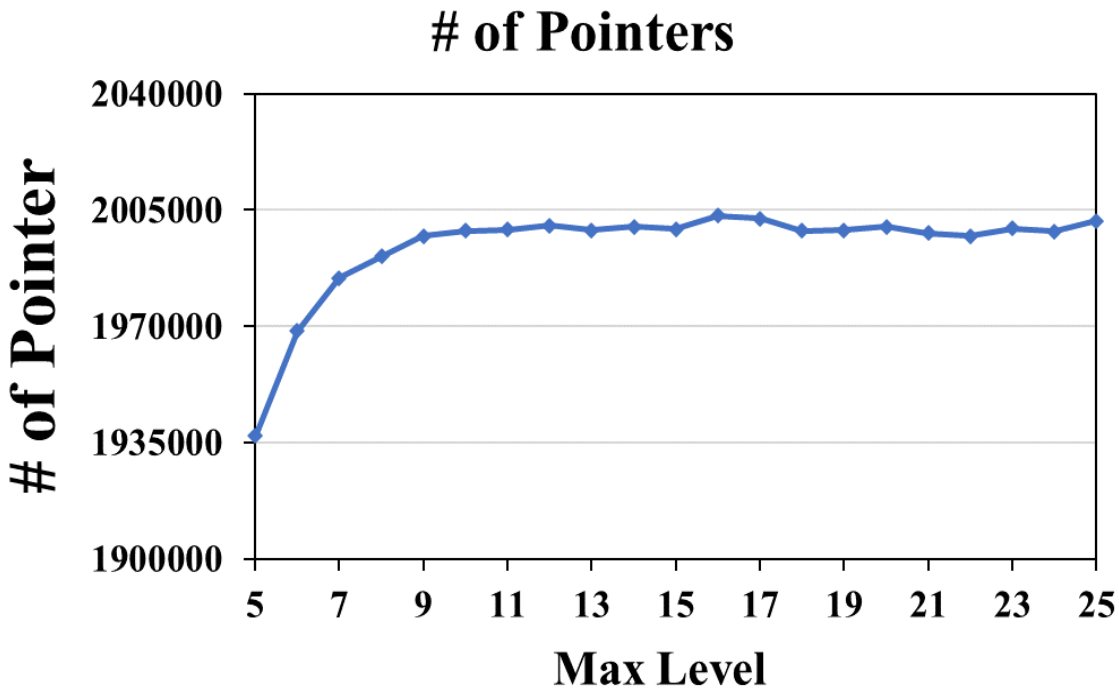




Effect of Max Level

- # of Elements: 1000000

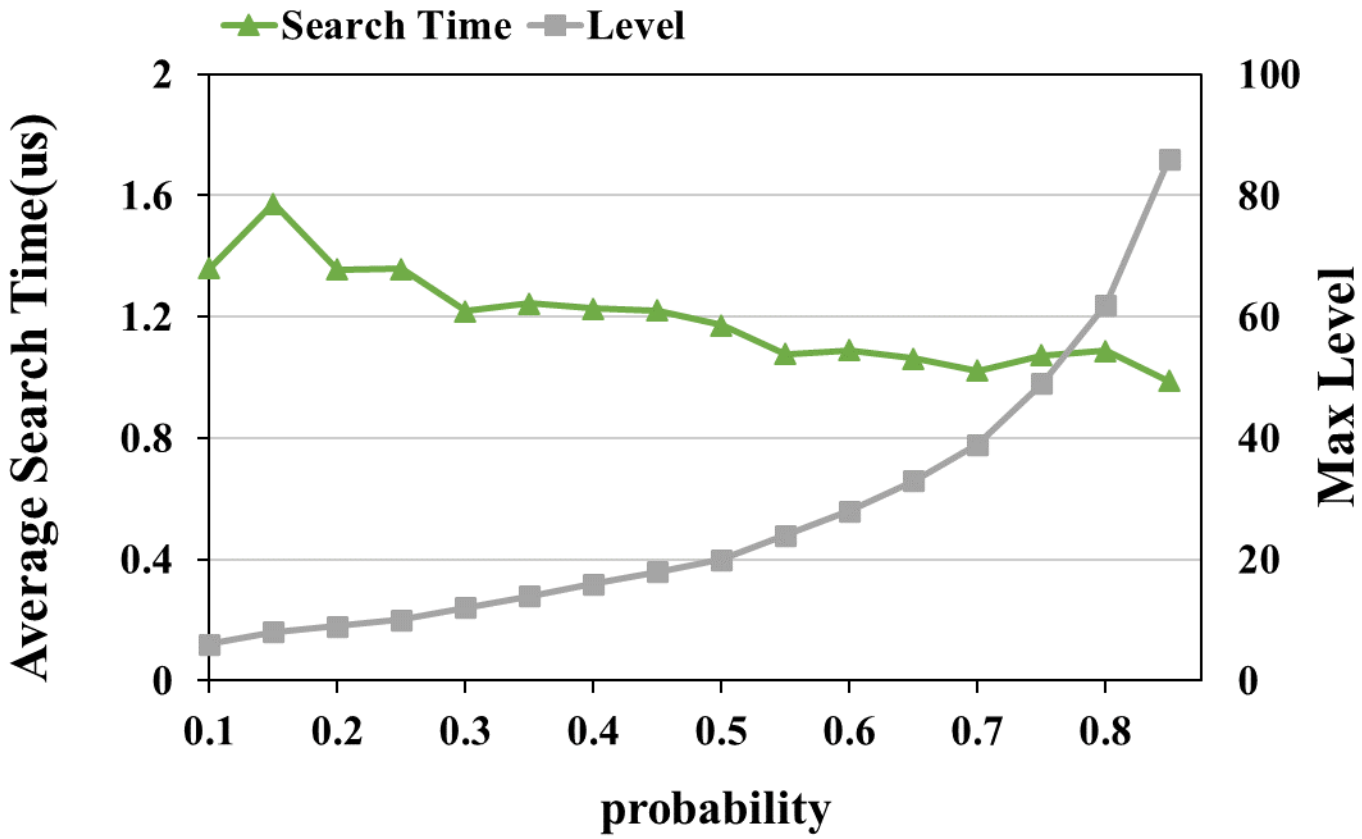




Default Max Level: $\log_{1/p} n = \log_{1/2} 1000000 \fallingdotseq 20$

Effect of Probability

- # of Elements: 1000000



Conclusion

- Skip-list와 Linked-list의 비교
  - 탐색 시간은 Skip-list가 더 빨랐지만 포인터의 개수가 2배정도 많았다.
- Skip-list의 파라미터를 변경하며 실험한 결과
  - 키의 개수가 많아지면 읽기 성능이 안 좋아진다.
  - 순차적인 쓰기가 쓰기 성능은 좋지만 읽기 성능은 안 좋다.
  - 최대 높이는 키의 개수에 따라 상한선이 있다.
  - 확률이 높아지면 최대 레벨이 급격히 높아진다.

Thank you

Q&A

2024.01.17

Presentation by Boseung Kim, Yeongyu Choi

bskim1102@dankook.ac.kr, dusrb1418@naver.com