

Tries

Jyh-Shing Roger Jang (張智星)
CSIE Dept, National Taiwan University

Intro to Tries

- An efficient data structure for **dictionary** (which uses strings as keys)
 - English dictionary (from a given word to its definition)
 - Yellow page (from a given name to his/her phone number)
- Why you need to learn tries
 - An alternative solution to `<map>` and `<unordered_map>`
 - For job interviews
 - For homework of IR

complexity in search

binary search: $O(m \log(n))$ m:length of key n:No. of key

Trie: $O(m)$ at space cost

ANIMATION OF TRIES

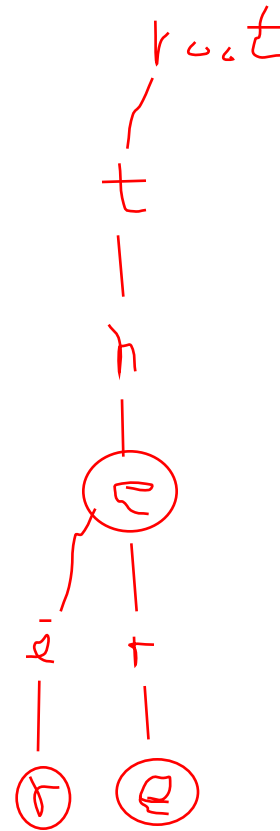
Quiz!

○ Construction

- key = the, a, there, answer, any, by, bye, their

○ Search

- any ✓
- these ✗



Code Examples

○ Examples

- trie00.cpp → C-flavored
- trie01.cpp → C++

Complexity

- Assuming we have n keys, each with m characters

- Tries

- Time complexity

Quiz!

- Construction: $O(m*n)$

- Search: $O(m)$

Space complexity: $O(\text{ALPHABET_SIZE} * m * n) = O(m * n)$

- ~~Space complexity: $O(m * n)$~~

- Binary search

- Time complexity

- Construction: $O(m * n * \log(n))$

- Search: $O(m * \log(n))$

- Space complexity: $O(m * n)$

References

- References to tries

- Video
 - [HackerRank](#)
 - [Trie data structure](#) (with detailed example)
 - CS50: [Kevin Schmid](#) , [Details](#)
- [Trie: Insert and search](#) (with code)
- [Programming tutorials by SourceTrick](#) (with code)
- CS50: [slides](#)