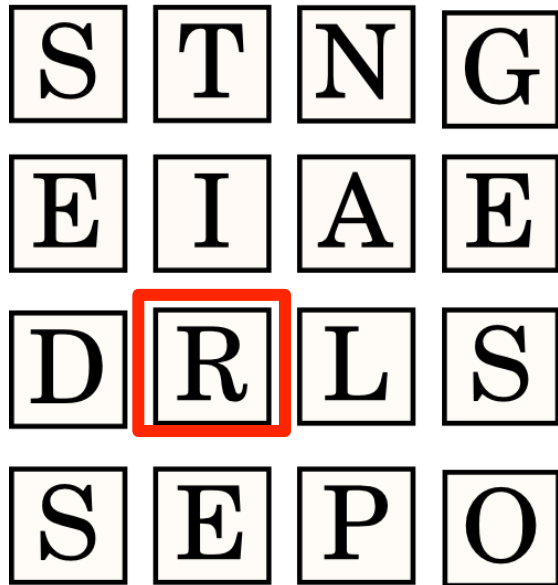


CSE 100 Lecture 21

Tries

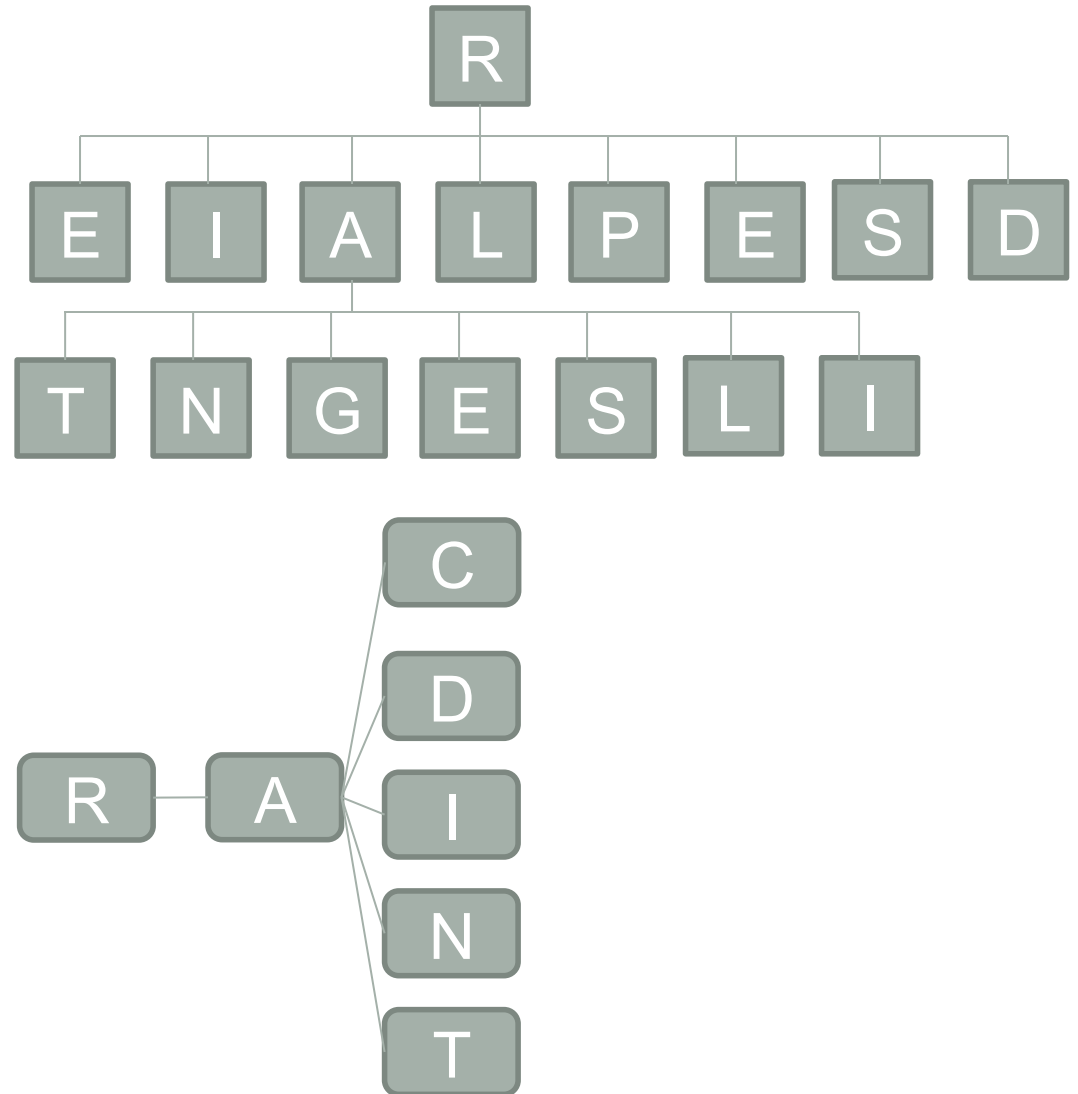
Two searches in PA4

Board



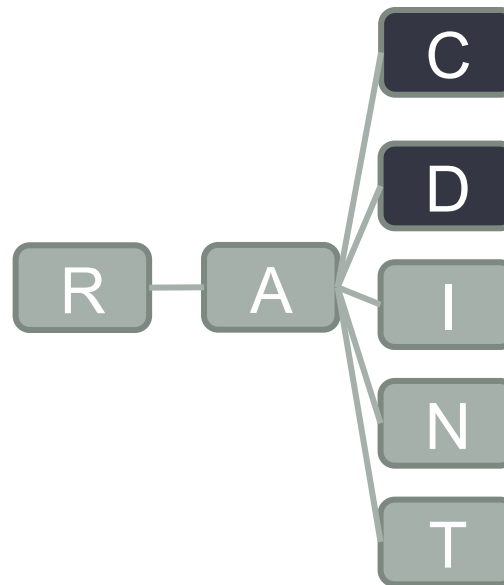
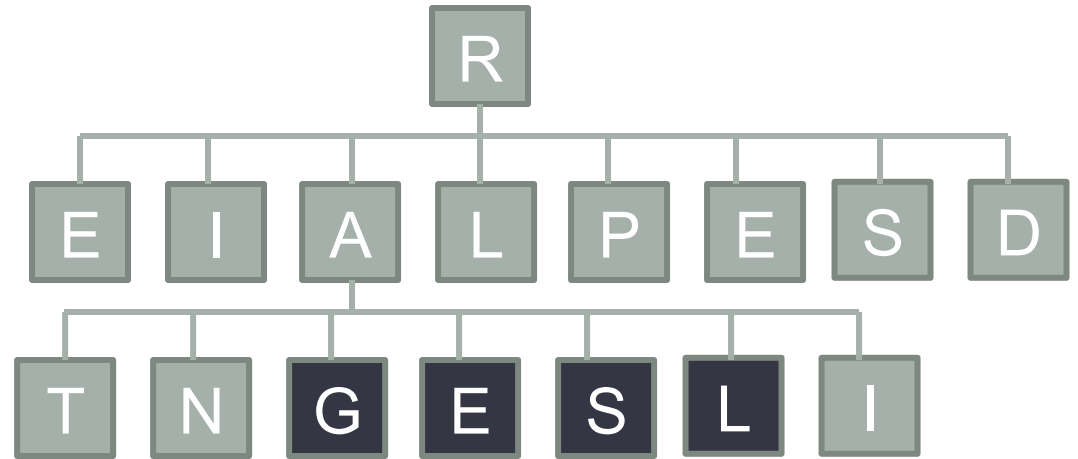
Dictionary

race
radiant
rain
ran
rat
rats
...



Two searches in PA4

Parallel search allows you to minimize time by ruling out unneeded paths. But how to represent the tree.



Tries: Efficient way to store/find keys that are sequences of digits

Create the code trie for the following code:

A: 00001

S: 10011

E: 00101

R: 10010

C: 0001

H: 101

Binary trie

A 00001

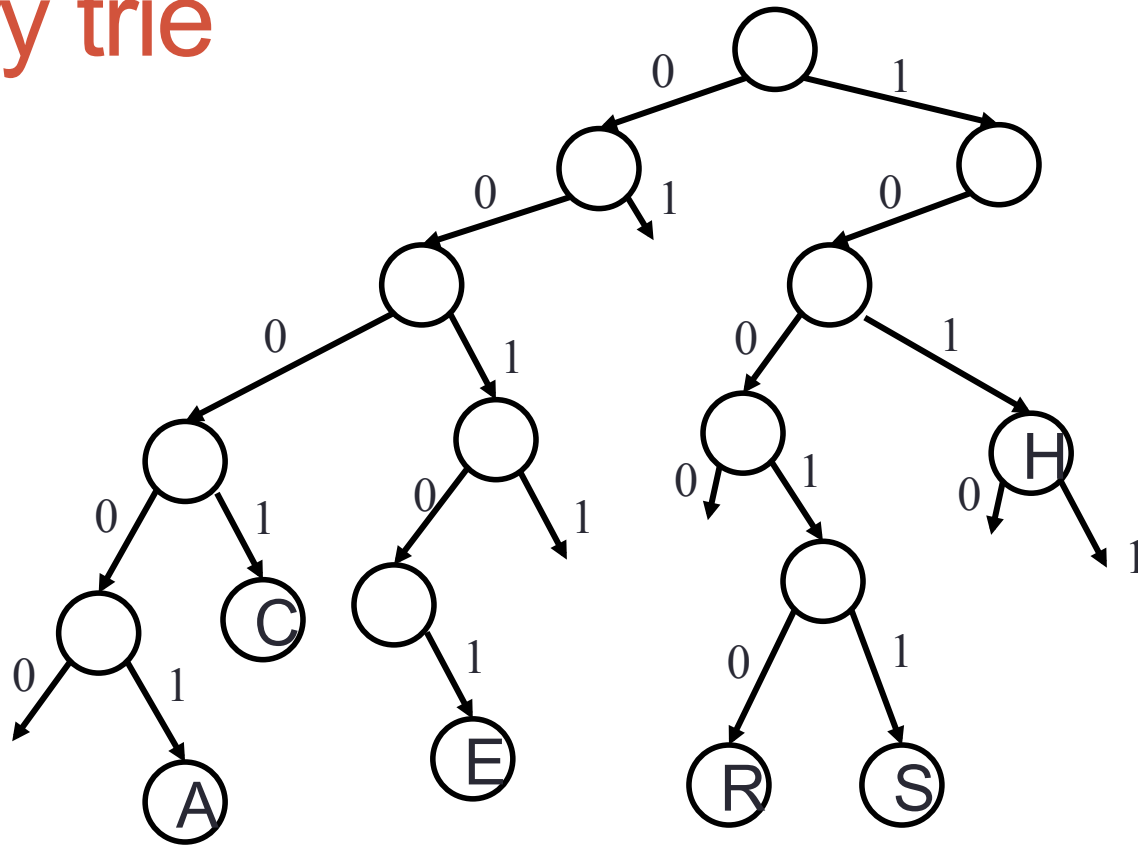
S 10011

E 00101

R 10010

C 0001

H 101



Does the structure of the trie depend on the order in which keys are inserted?

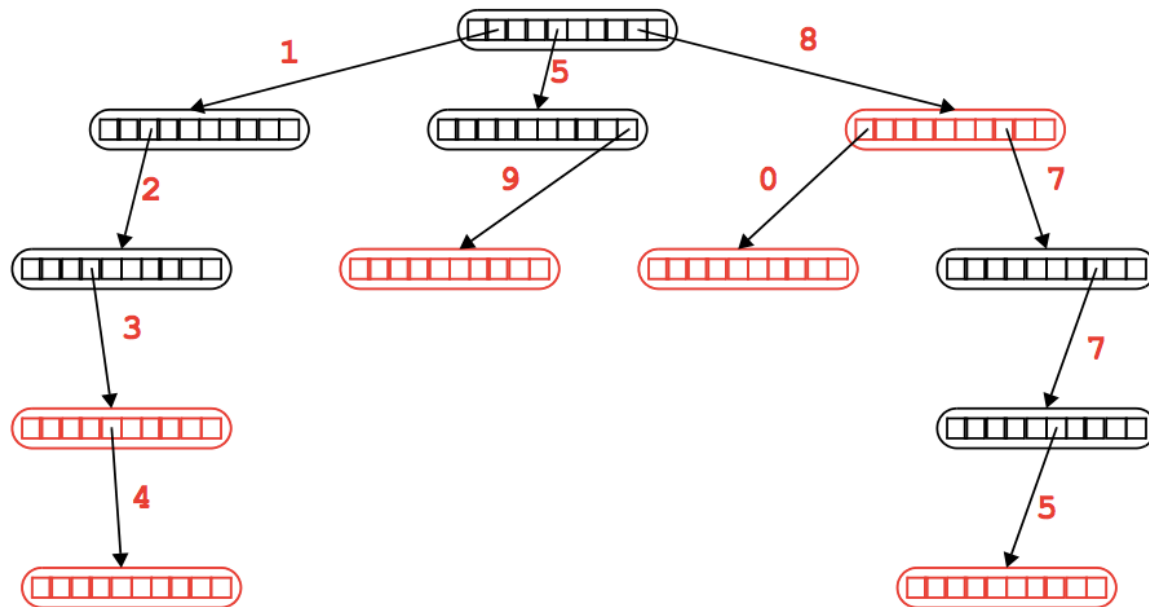
A. Yes

B. No

Multi-way tries: Efficient finding of keys by their sequence

Build the trie which holds the following number keys:

8
1234
59
123
8775
80



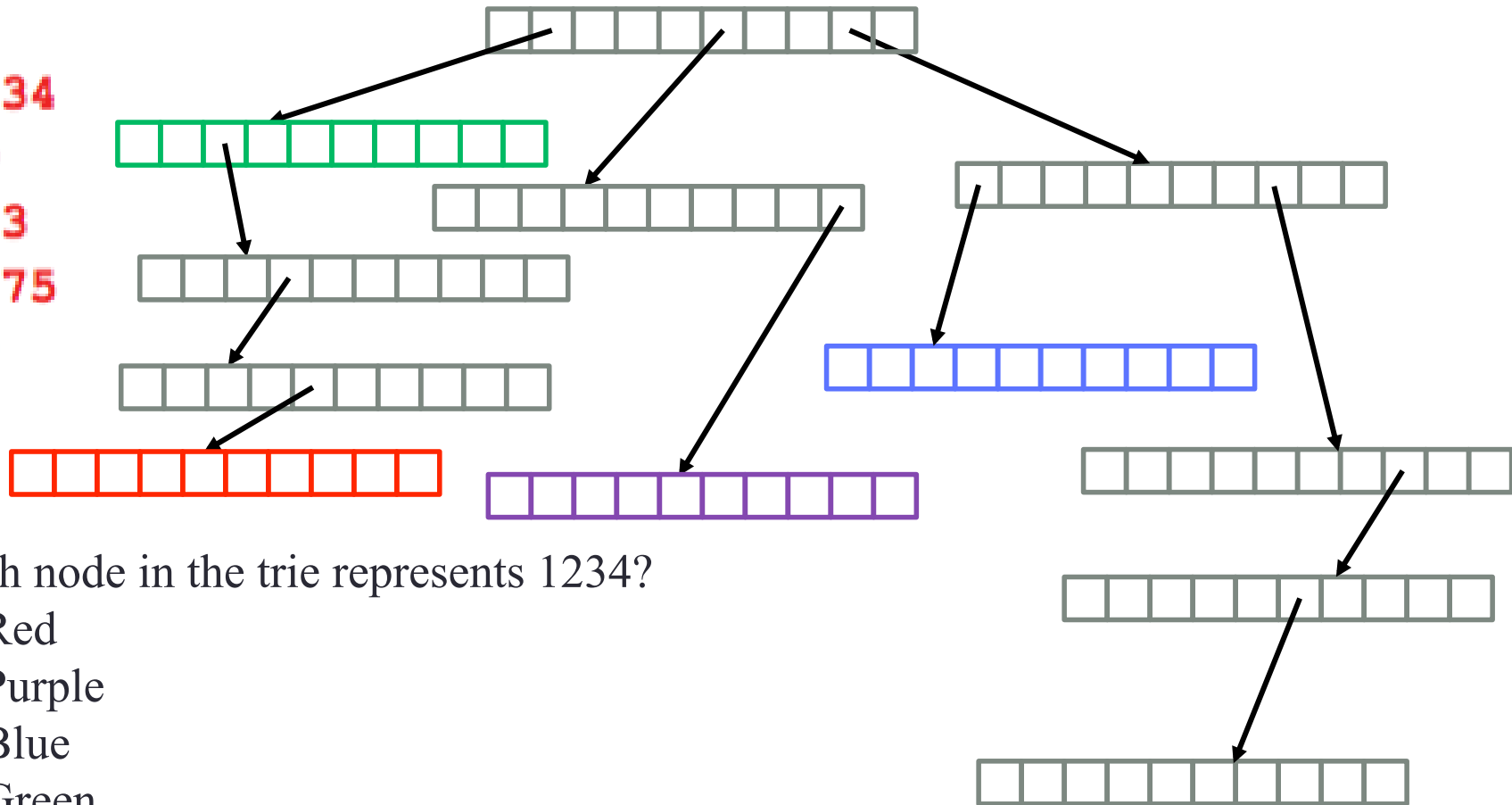
Assuming your trie could potentially hold *any* decimal number,
How many children does each node (potentially) have?

- A. 2 B. 8 C. 10 D. Other

Multi-way tries: Efficient finding of keys by their sequence

Build the trie to store the following numbers:

8
1234
59
123
8775
80



Which node in the trie represents 1234?

- A. Red
- B. Purple
- C. Blue
- D. Green
- E. Other

Multi-way tries: Efficient finding of keys by their sequence

Build the trie to store the following numbers:

8

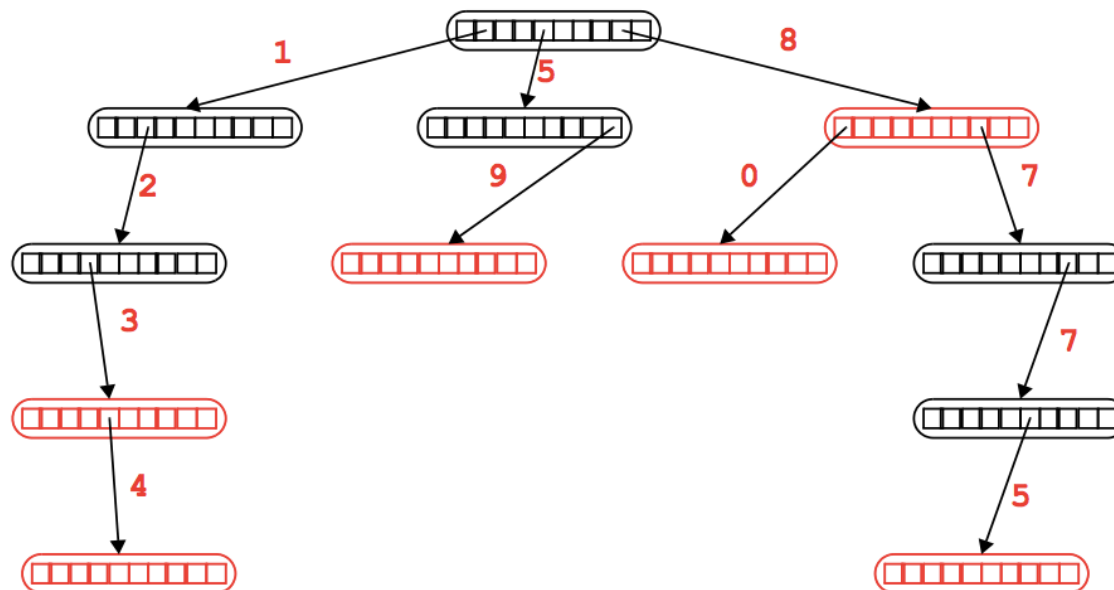
1234

59

123

8775

80



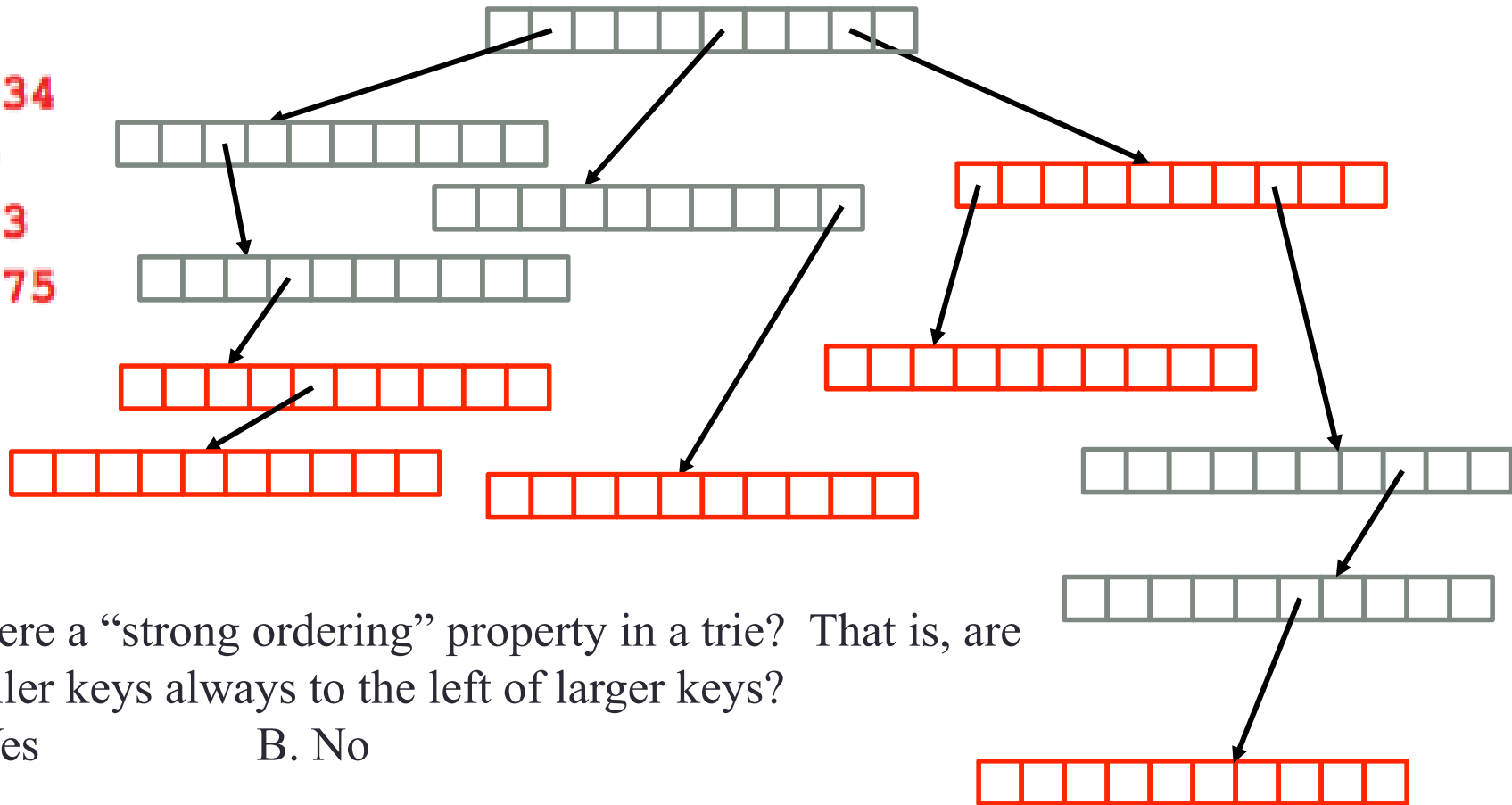
Is there a way to find whether *all* keys contained in a sequence of digits are present in the trie?

- A. Yes
- B. No

Properties of tries

Build the trie to store the following numbers:

8
1234
59
123
8775
80



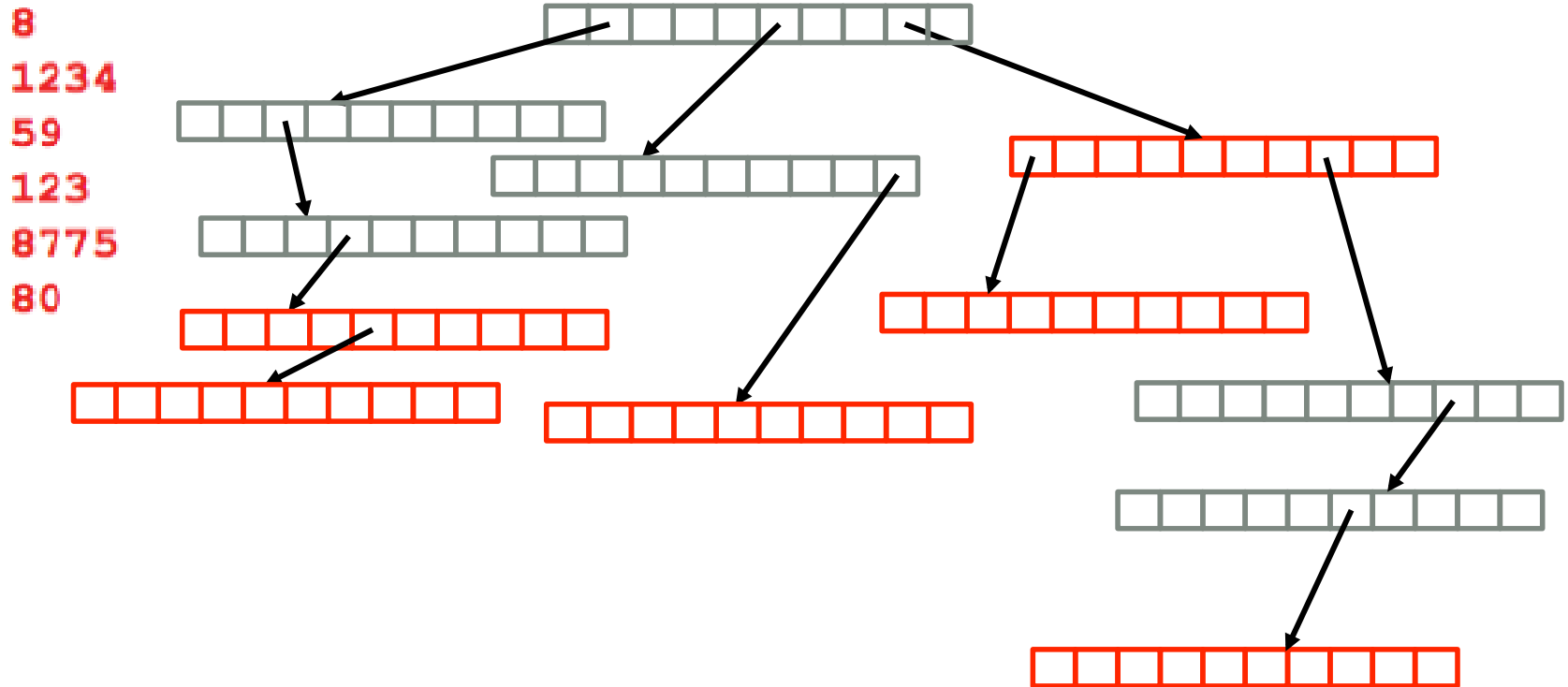
Is there a “strong ordering” property in a trie? That is, are smaller keys always to the left of larger keys?

A. Yes

B. No

Properties of tries

Build the trie to store the following numbers:

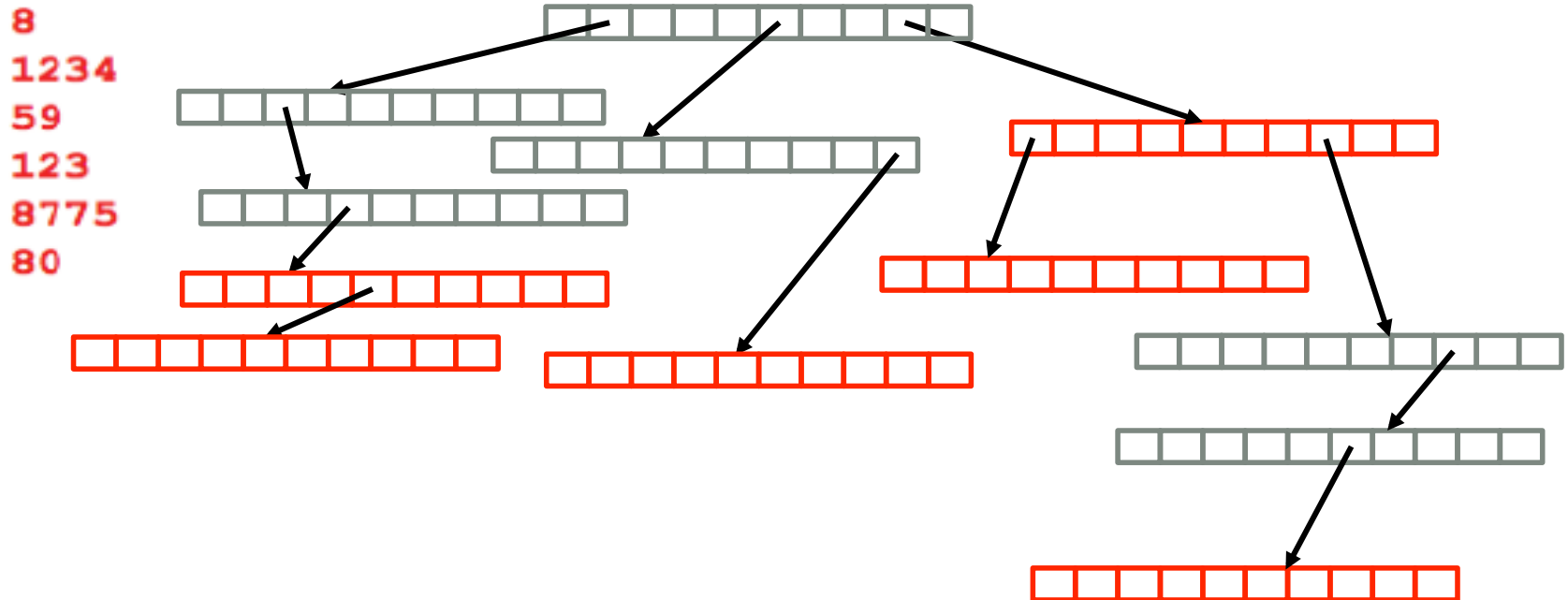


Suppose your keys are a sequence of at most D digits, and N is the maximum number of keys you will store. What is the worst case height of this trie (i.e., for large N)?

- A. D B. $\lg(D)$ C. N D. $D \times N$ E. Other

Properties of tries

Build the trie to store the following numbers:

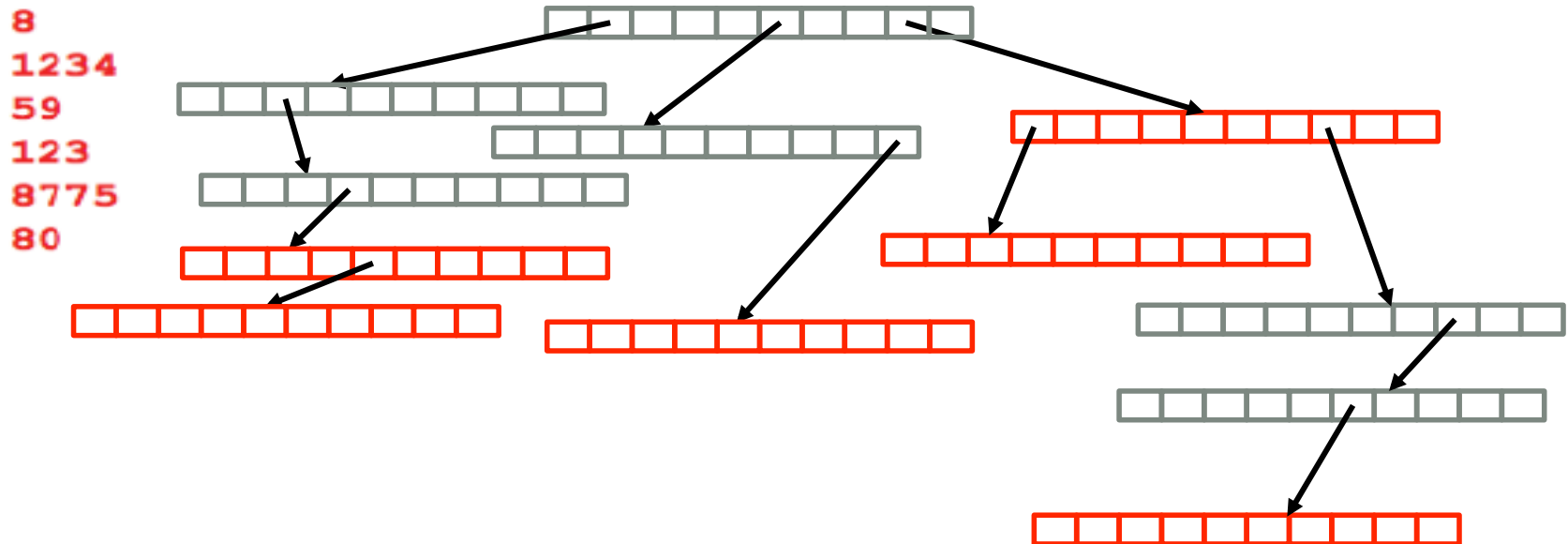


If you stored the same N D -digit keys in a Binary Search Tree, what would be the worst case height of the tree?

- A. N B. $\lg(10^D)$ C. $\lg(N)$ D. $\lg(D)$ E. Other

Properties of tries

Build the trie to store the following numbers:



Consider storing the full 10^D keys. We know that on average the height of a BST will be $\lg(10^D)$. Which is smaller: D or $\lg(10^D)$?

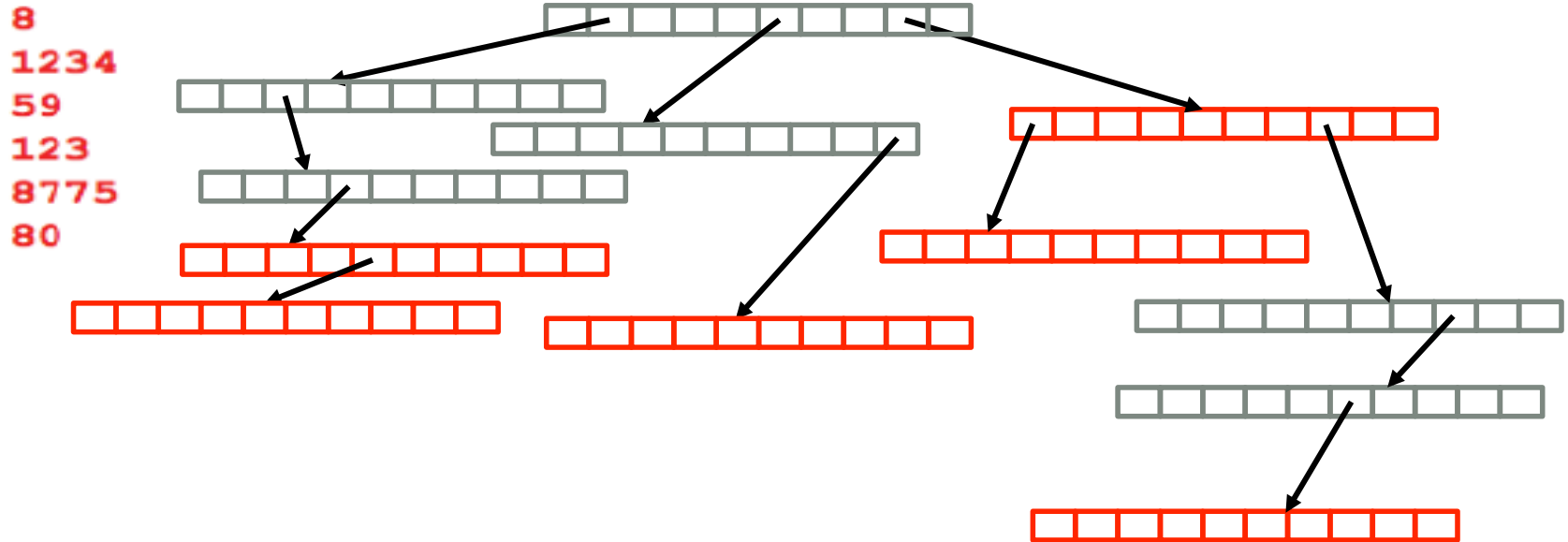
A. D

B. $\lg(10^D)$

C. They are the same

Properties of tries

Build the trie to store the following numbers:



So what is the main drawback of tries?

- A. They are difficult to implement
- B. They (usually) waste a lot of space
- C. They are slow
- D. There is no drawback of tries