Tries and More Graphs

Discussion 11

CS 61B // Spring 2021

Announcements

Week 11

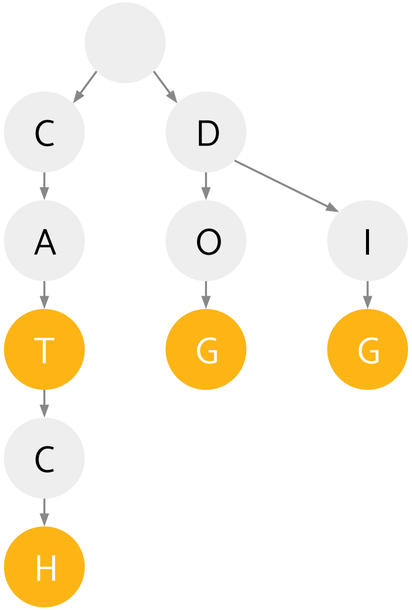
- ❑ Week 11 Survey due Tuesday, 4/6
- ❑ Lab 12 due Friday, 4/9
- ❑ Lab 13 due Friday 4/16
- ❑ Project 3 Phase 1: World Generation due Friday, 4/16
- ❑ Project 3 Phase 2: Interactivity due Thursday, 4/27 (no slip days)
- ❑ Final Exam is Tuesday, 5/11 8-11AM PT

Content Review

Tries

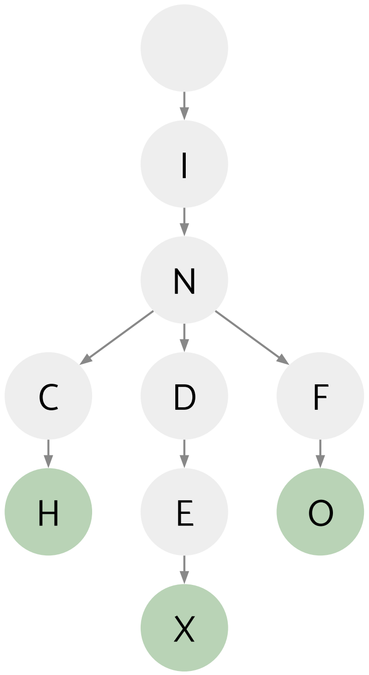
Tries are special trees mostly used for language tasks.

Each node in a trie is marked as being a word-end or not, so you can quickly check whether a word exists within your structure.



Worksheet

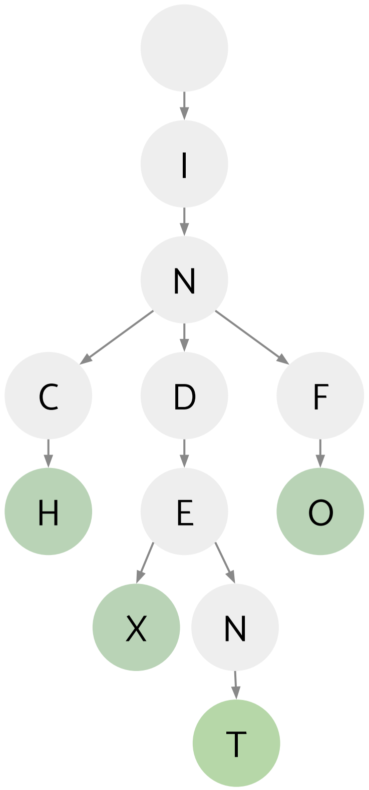
1A Trie Your Best



What strings are stored in the trie shown?

Now insert the strings "*indent*", "*inches*", and "*trie*" into the trie.

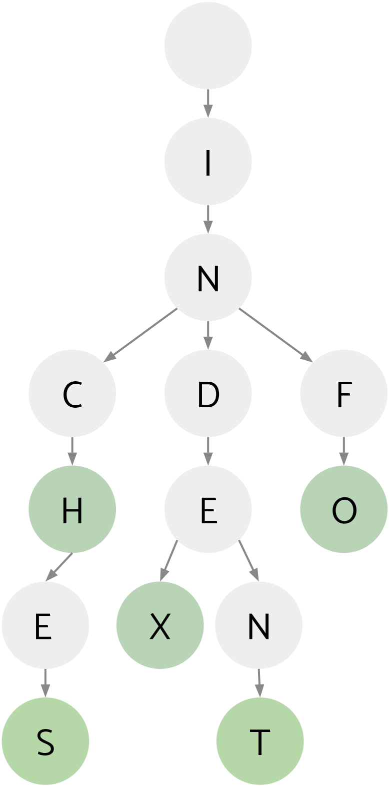
1A Trie Your Best



What strings are stored in the trie shown?

Now insert the strings “***indent***”, *“inches”*, and “*trie”* into the trie.

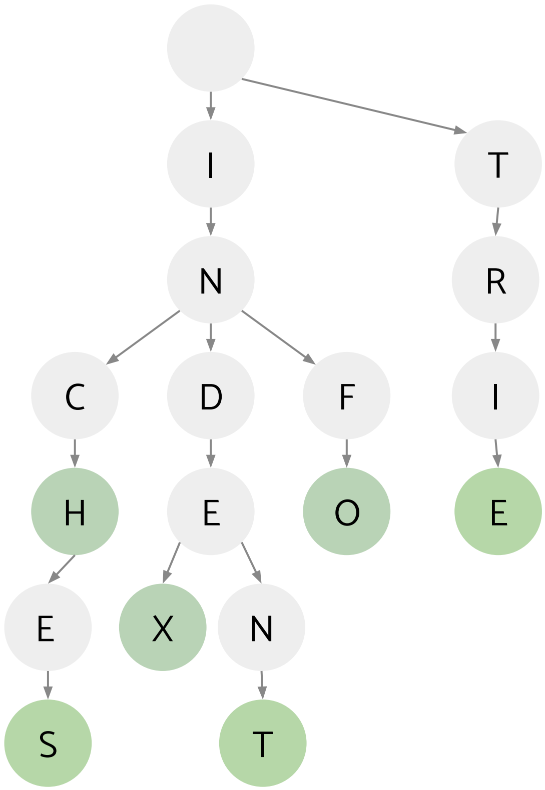
1A Trie Your Best



What strings are stored in the trie shown?

Now insert the strings "*indent*", ***"inches"***, and "*trie*" into the trie.

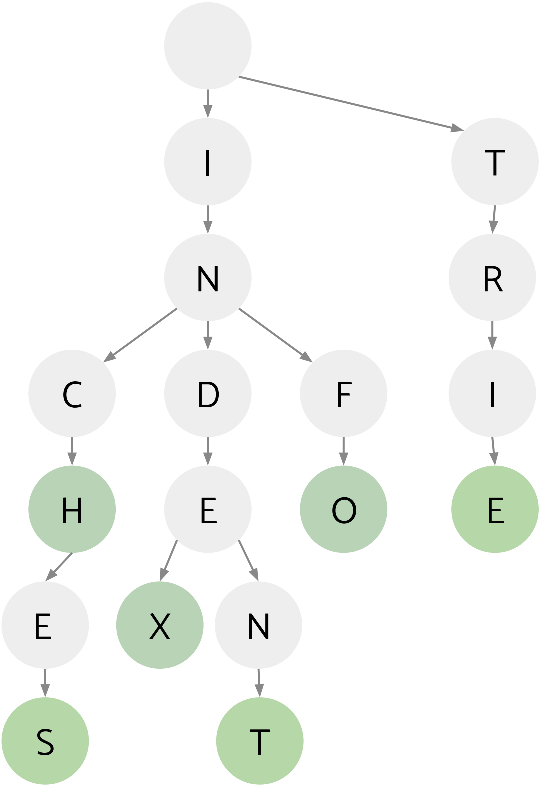
1A Trie Your Best



What strings are stored in the trie shown?

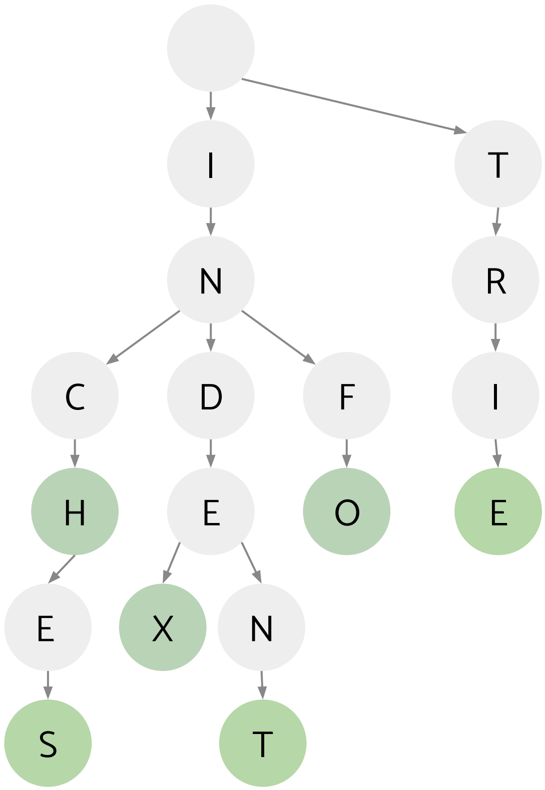
Now insert the strings “*indent*”, “*inches*”, and “***trie***” into the trie.

1B Trie Your Best



What is the runtime to find out if a given string is in the trie? What is the runtime to add a string to the trie? Describe your answers in terms of N, the number of words in the trie. You may assume the max length of any word in the trie is a constant.

1C Trie Your Best *Extra*



How could you modify a trie so that you can efficiently determine the number of words with a specific prefix in the trie? Describe the runtime of your solution.

2 A Tree Takes on Graphs

Your friend at Stanford has made some sus statements about graphs–you're pretty sure they're all false but you want to help your friend see why.

For each of the following parts, provide a counterexample that disproves their false statement.

2A A Tree Takes on Graphs

“Every graph has one unique MST.”

2B A Tree Takes on Graphs

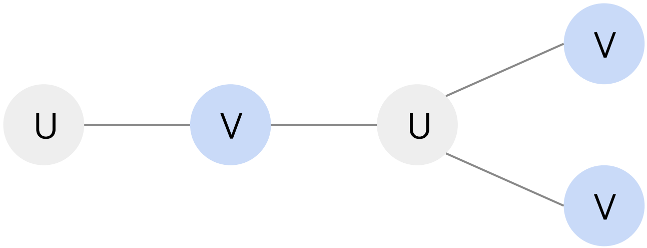
"No matter what heuristic you use, A* search will always find the correct shortest path."

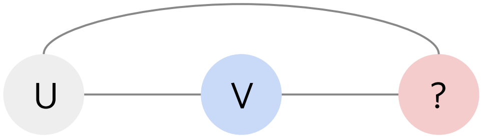
2C A Tree Takes on Graphs

”If you add a constant factor to each edge in a graph, Dijkstra’s algorithm will return the same shortest paths tree.”

3A Graph Algorithm Design

An undirected graph is said to be bipartite if all of its vertices can be divided into two disjoint sets *U* and *V* such that every edge connects an item in *U* to an item in *V*. For example below, the graph on the left is bipartite, whereas on the graph on the right is not. Provide an algorithm which determines whether or not a graph is bipartite. What is the runtime of your algorithm? *Hint: Can you modify an algorithm we already know?*





3B Graph Algorithm Design

Consider the following implementation of DFS, which contains an error:

```
create the fringe, which is an empty Stack
push the start vertex onto the fringe and mark it
while the fringe is not empty:
    pop a vertex off the fringe and visit it
    for each neighbor of the vertex:
        if neighbor not marked:
            push neighbor onto the fringe
            mark neighbor
```

Identify the bug, then give an example of a graph where this algorithm may not traverse in DFS order.

3B Graph Algorithm Design

Consider the following implementation of DFS, which contains an error:

```
create the fringe, which is an empty Stack
push the start vertex onto the fringe and mark it
while the fringe is not empty:
    pop a vertex off the fringe and visit it
    for each neighbor of the vertex:
        if neighbor not marked:
            push neighbor onto the fringe
            mark neighbor
```

Identify the bug, then give an example of a graph where this algorithm may not traverse in DFS order.

3B Graph Algorithm Design

Consider the following implementation of DFS, which contains an error:

```
create the fringe, which is an empty Stack
push the start vertex onto the fringe and mark it
while the fringe is not empty:
    pop a vertex off the fringe and visit it
    if the vertex was not already marked:
        mark the vertex you just popped
        for each neighbor of the vertex:
            if neighbor not marked:
                push neighbor onto the fringe
```

Identify the bug, then give an example of a graph where this algorithm may not traverse in DFS order.

3B Graph Algorithm Design

Consider the following implementation of DFS, which contains an error:

```
create the fringe, which is an empty Stack
push the start vertex onto the fringe and mark it
while the fringe is not empty:
    pop a vertex off the fringe and visit it
    for each neighbor of the vertex:
        if neighbor not marked:
            push neighbor onto the fringe
            mark neighbor
```

Identify the bug, **then give an example of a graph where this algorithm may not traverse in DFS order.**

3C Graph Algorithm Design *Extra*

Provide an algorithm that finds the shortest cycle (in terms of the number of edges used) in a directed graph in O(EV) time and O(E) space, assuming E > V.