

Boggle

and more recursive backtracking

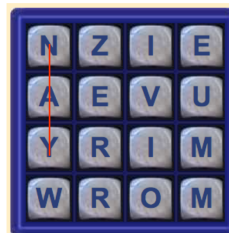
Today

- Intro to Boggle
 - Trees and recursive backtracking!!!!
- Practice with recursive backtracking

How to play



How to play



How to find words?



- You have a dictionary
- You want to find all words on board
- How?
- Write out step-by-step algorithm

Brute force



- Find everything!
- N
 - N-Z
 - N-Z-I
 - N-Z-V
 - N-Z-E
 - N-Z-A
 - N-E
 - N-A

Is there a better way?



- ~~Find everything!~~
- N
 - N-Z
 - N-Z-I
 - N-Z-V
 - N-Z-E
 - N-Z-A
 - N-E
 - N-A

Is there a better way?

- ~~Find everything!~~
- N



Words that start with Nz

A list of words that start with Nz (words with the prefix Nz). We search the official scrabble dictionary for scrabble words starting with nz – we take the letter or word you enter, and generate all words starting with Nz. In addition there is a list of [Words that end with nz](#), [words that contain nz](#).

Search for words that start with a letter or word:

No Words Found! Try words that end with nz or words that contain nz instead?

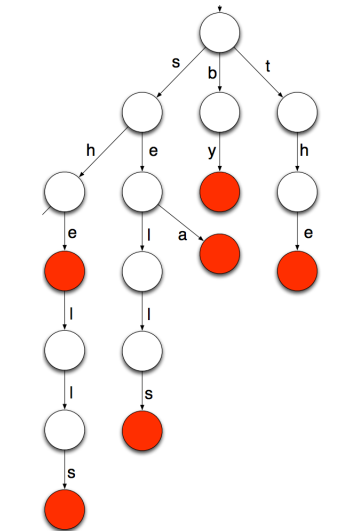
Prefixes



- Look at word prefixes
- If prefix doesn't make a word, STOP
- We should store our words in a data structure to help us

Tries

- Tree-like structure
 - Node
 - word prefix/end
 - Pointer to # nodes
 - Pointer has letter





Let's try it!

- “Peter piper picked a peck of pickled peppers”



Today

- Intro to Boggle
 - Trees and recursive backtracking!!!!
- Practice with recursive backtracking



Recursion

- `void solve(ProblemClass instance){`
 - base cases
 - recursion
 - reassemble problem
- `}`



Recursive backtracking

- `void solve(ProblemClass instance){`
 - base cases
 - save current state
 - recursion
 - if current state DIDN'T work
 - remove current state
 - reassemble problem
- `}`

Recursive backtracking

- Write *getPathSum*
 - Takes int as *target* sum
 - Returns Stack of Intgers, the path to target sum
 - *getPathSum(19)* returns 7, 15
 - *getPathSum(8)* return an empty stack

