CS 350 2024-25 Sem I

Lecture 5

Aug 10, 2024

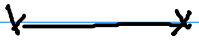
Recall:

Higher order functions

```
map    :: (a -> b) -> [a] -> [b]
filter :: (a -> Bool) -> [a] -> [a]
foldr  :: (a -> b -> b) -> b -> [a] -> b
```

How to write functions using higher order functions to avoid iterative style programming

- primes up to n
- base 60



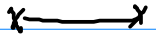
Package manager: cabal

Package site: hackage

google for haskell packages

stackoverflow

reddit r/haskell



Puzzle

"food" ⟶ "Dino"

in a sequence of transition words, where adjacent words differ by exactly one character.

① Read words from a word list

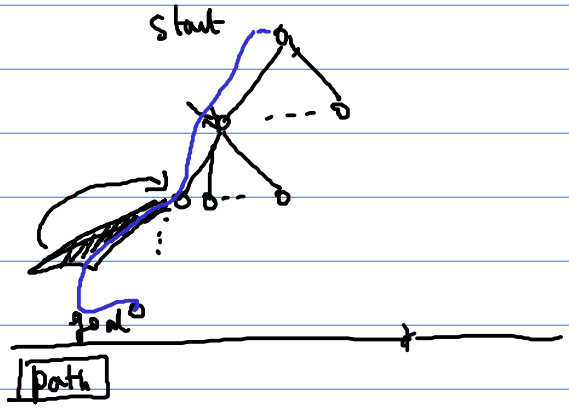
② Do a dfs starting from the given word, until we reach the goal.

"food"
"Dino"

nodes = all words of equal length

edge = (w1, w2) has an edge if w1 & w2 are at distance 1 from each other

DFS



dfs path nodes

for v in neighbors:

l' ← dfs (v: path)
nodes \\[v]
goal

If (head l') == goal
then l'
otherwise continue to next node in neighbors.

List comprehension is iterative.

[x * x | x ← [1 .. n]]

[☐ | v ← neighbors]