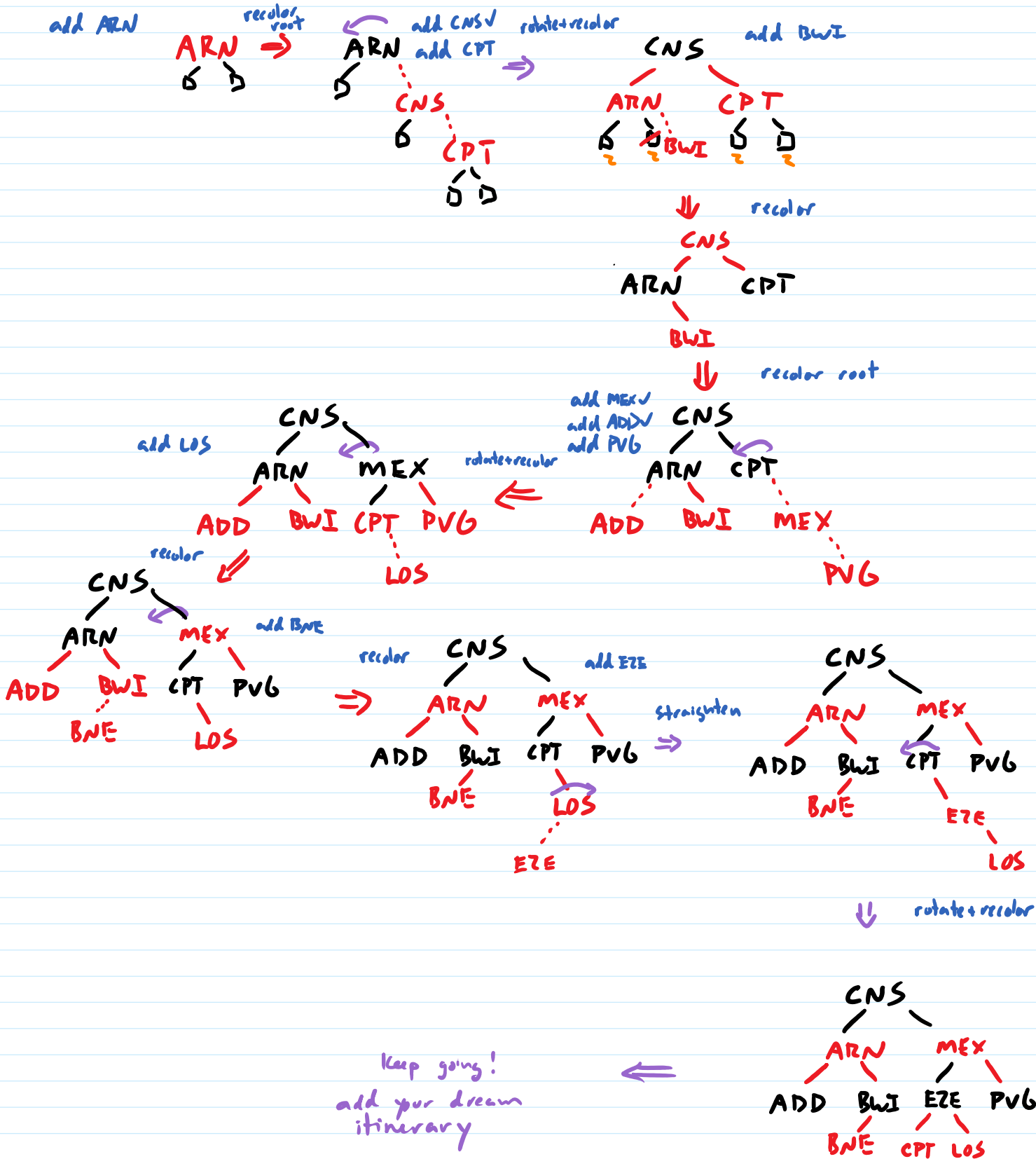


Example

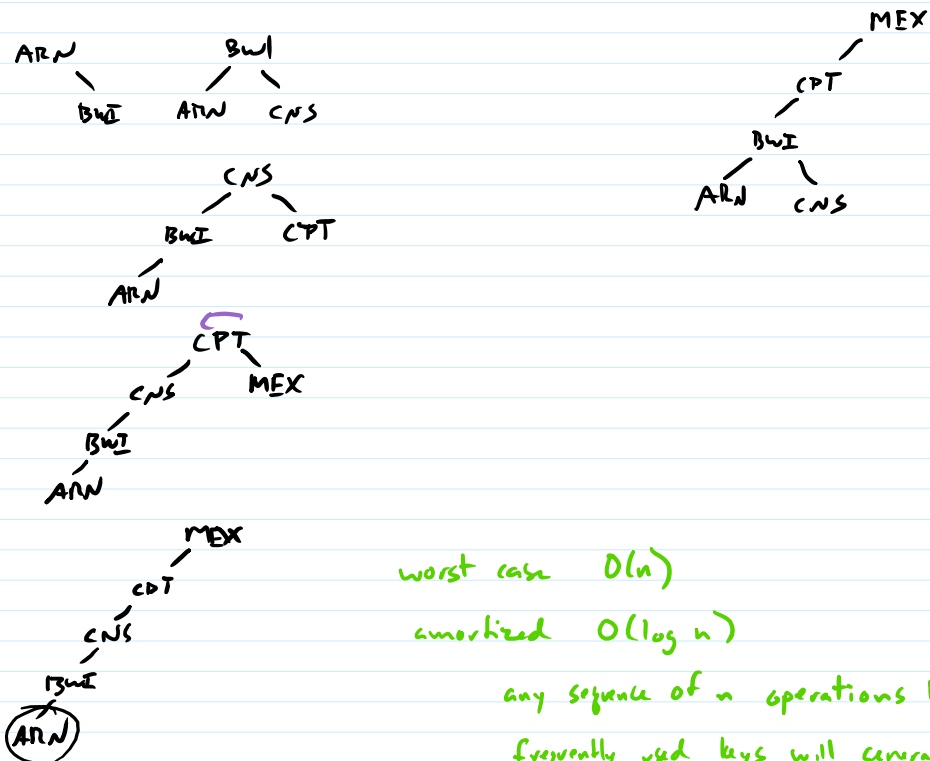
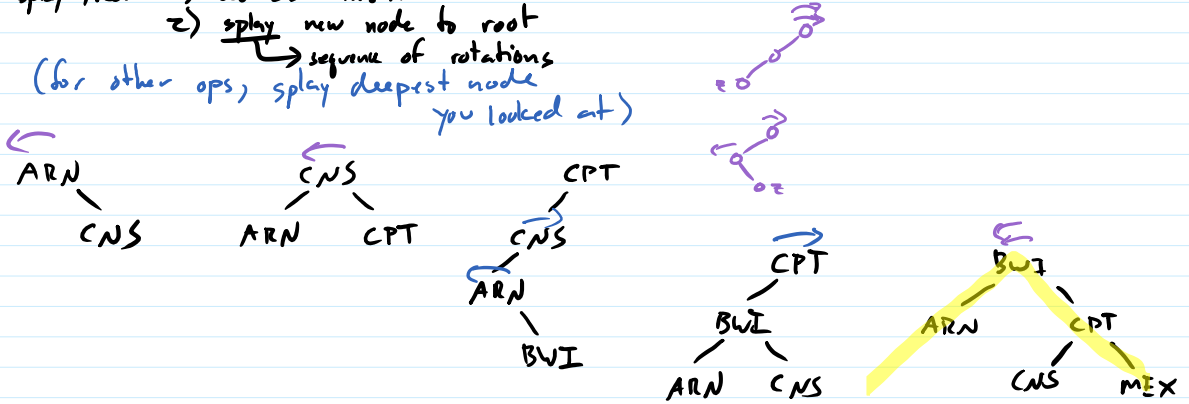
ARN CNS CPT BWI MEX ADD PVG LOS BNE EZE



Keep going!  
 add your dream  
 itinerary  
 KOA HND ccv....

ARN CNS CPT BWI MEX ADD PVG LOS BNE EZE

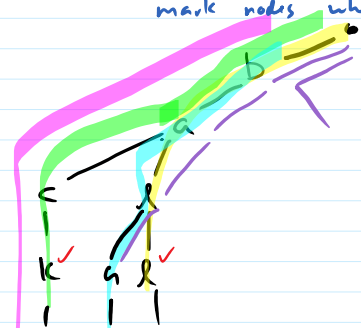
insert into splay tree: 1) do BST insert  
 2) splay new node to root  
 (for other ops, splay deepest node you looked at)  
 → sequence of rotations



worst case  $O(n)$   
 amortized  $O(\log n)$   
 any sequence of  $n$  operations takes  $O(n \log n)$   
 frequently used keys will generally be close to top

tRie - one child per possible next letter  
 mark nodes where words end ✓

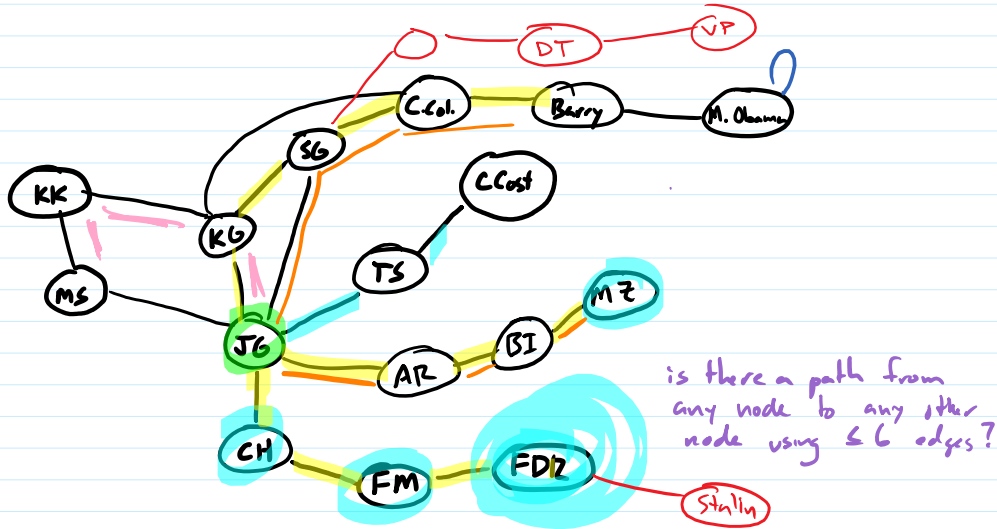
ball  
 ball



can determine if a word is in the set in  $O(k)$  time where  $k = \text{length of word}$

k ✓  
i ✓  
n ✓  
s ✓  
a ✓  
r ✓  
c ✓  
e ✓  
l ✓  
e ✓  
r ✓

Graphs



Graph: represents things and relationships between them  
 nodes people  
 edges relationship

path: sequence of vertices w/ edges between JG KG KK MS JG TS CC

simple path: no repeated vertices JG KG KK MS

cycle: starts/ends @ same place JG KG KK MS JG KG KK MS JG

simple cycle: no repeats except beginning/end

```

int foo(int n, int c)
{
  if (n == c)
  {
    return 0;
  }
  int i = 1;
  while (i < n)
  {
    if (i % c == 3)
    {
      if (n % 2 == 1)
      {
        return 0;
      }
    }
    i++;
  }
}

```

vertices: lines of code (node)

edges: control flow

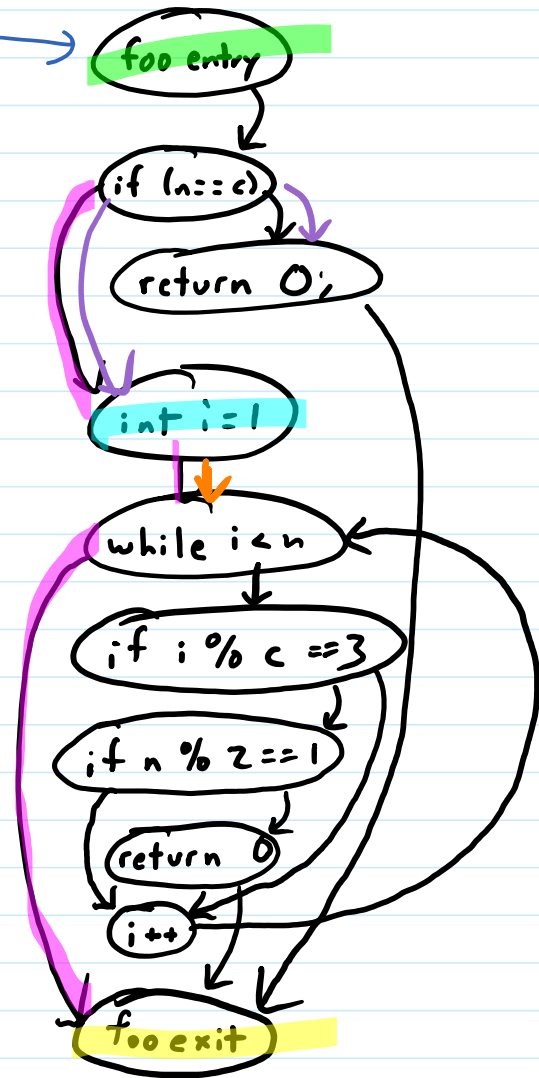
$u \rightarrow v$   
means  $v$  can follow  $u$

is there a path

entry  $\rightarrow$  exit

that doesn't hit a return

yes

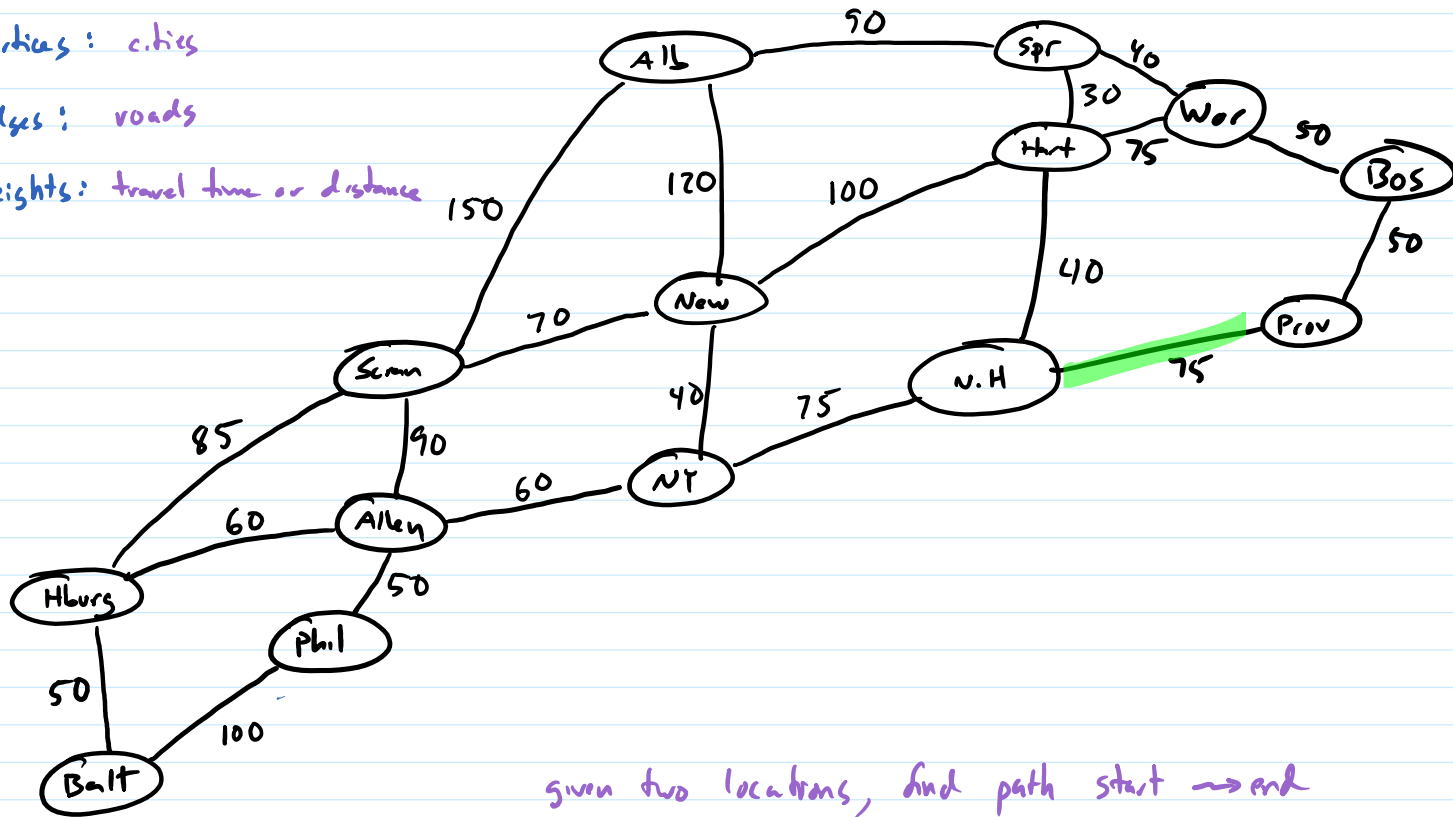


# Shortest Path

vertices: cities

edges: roads

weights: travel time or distance



given two locations, find path start  $\rightarrow$  end  
w/ lowest total weight

vertices: intersections

edges: segments of roads

weights: travel time / distance

