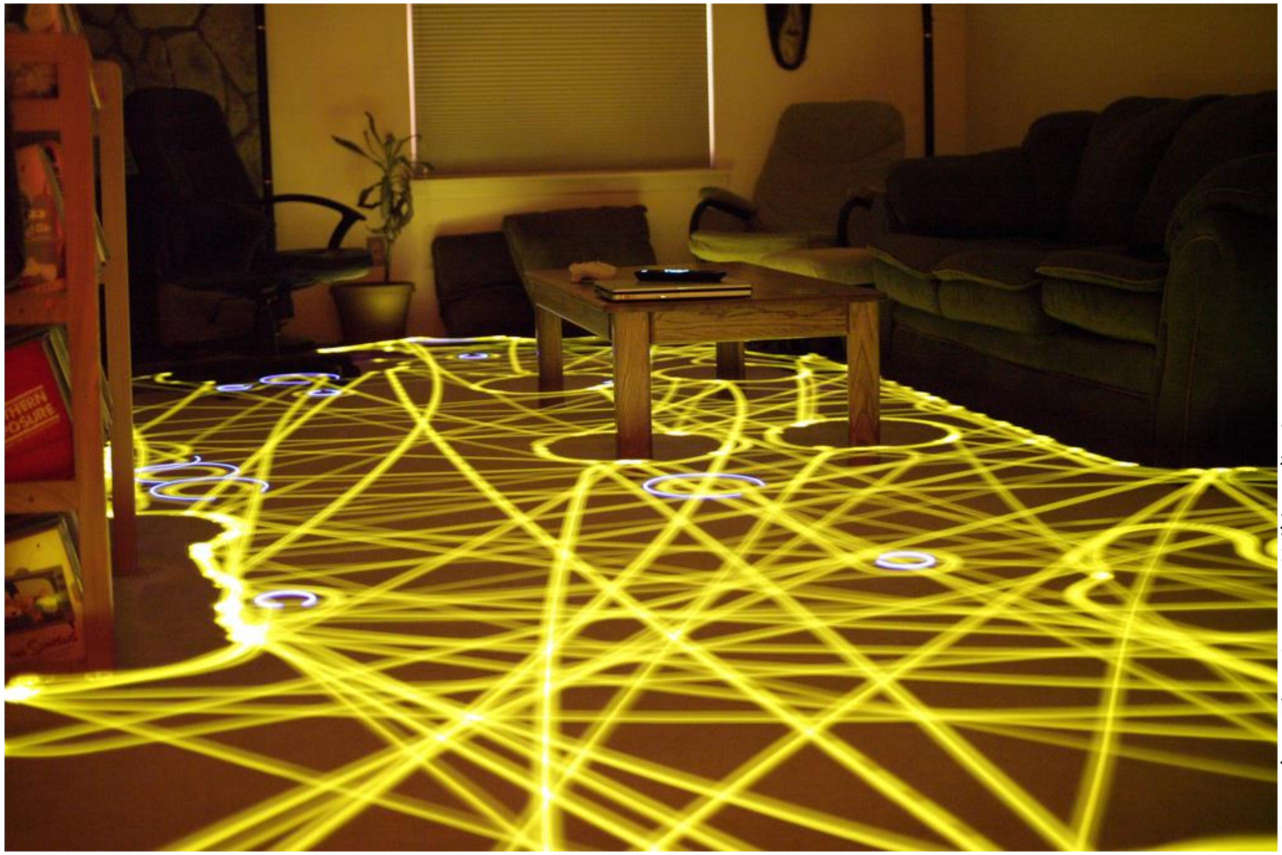
e yantra

Robotics Competition Plus Pilot



Roomba - Time Lapse – Wikimedia Commons

Path Planning for Robots

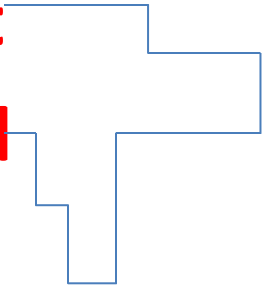
Tutorial

Introduction

A path is a way of connecting two points,

a **start**

an **end**



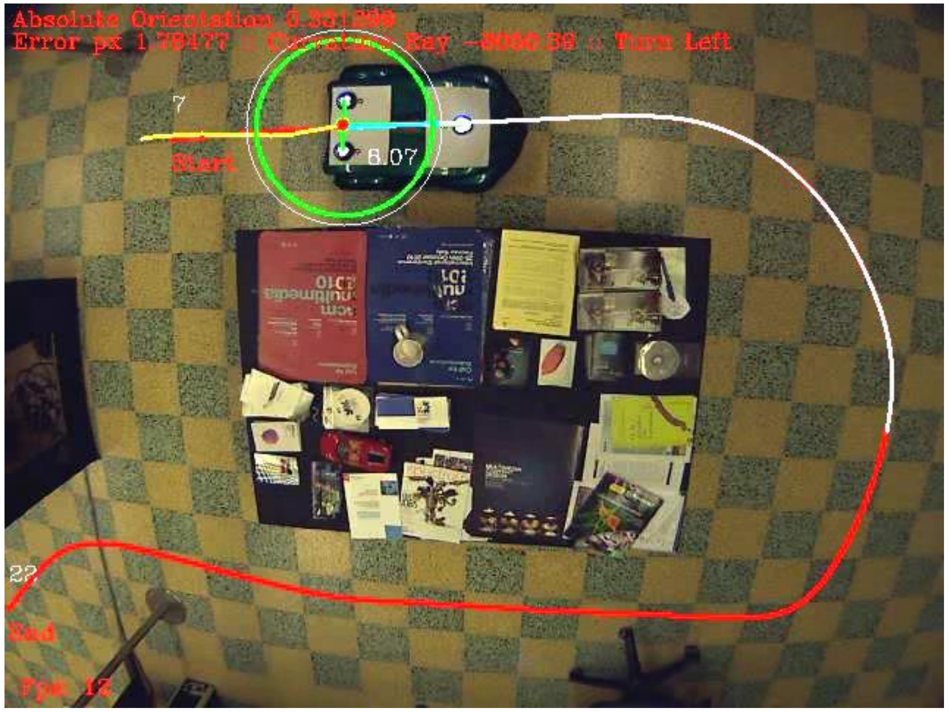
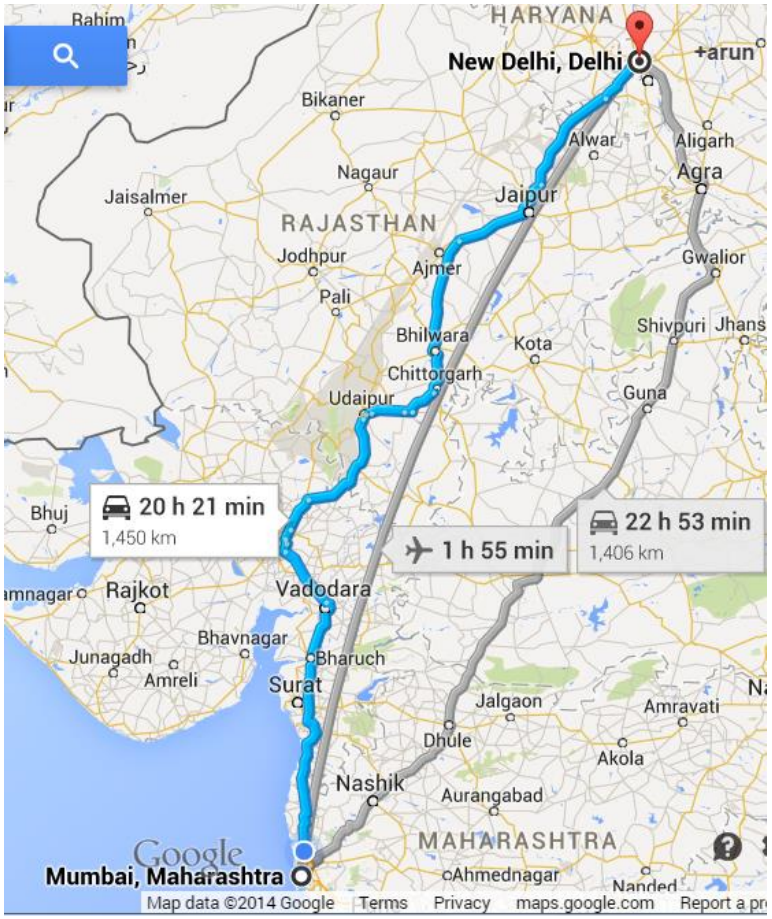

Absolute Orientation 0.331299
Error px 1.79477 :: Curvature Ray -8080.39 :: Turn Left
7
8.07
Start
22
End
Fps 12

Image taken from http://www.micc.unifi.it/vim/projects/wheeled-mobile-robot-path-follower-in-uncalibrated-multiple-camera-environment/

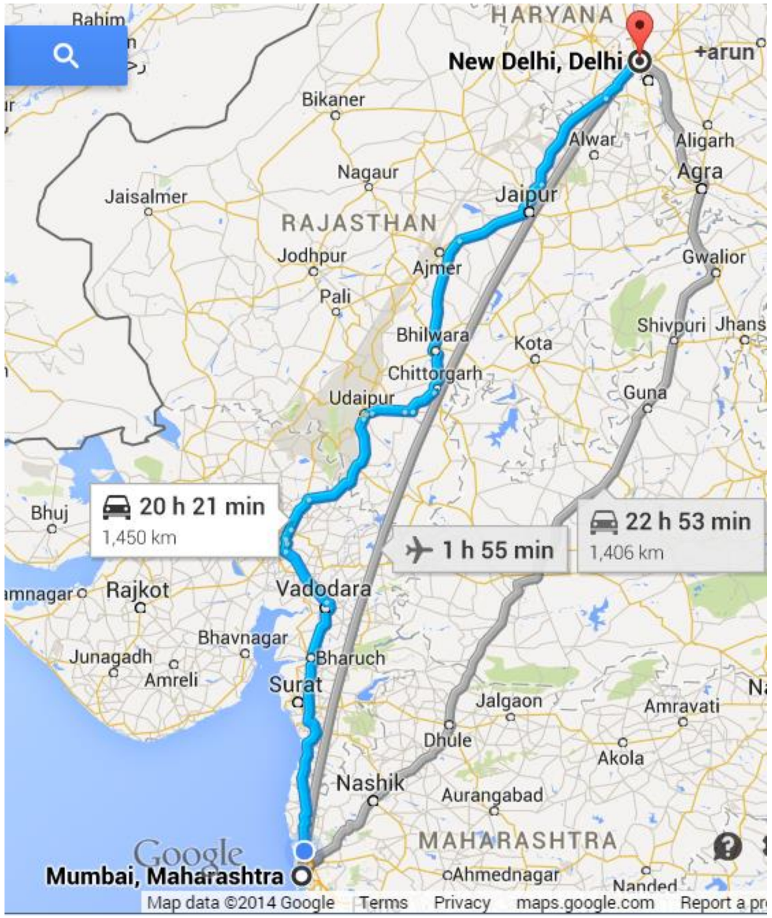
Path Planning

- How can paths be found?
- What happens in case there are many paths?
- How can we tell which path is the better one?
- For example, consider the path from Mumbai to Delhi



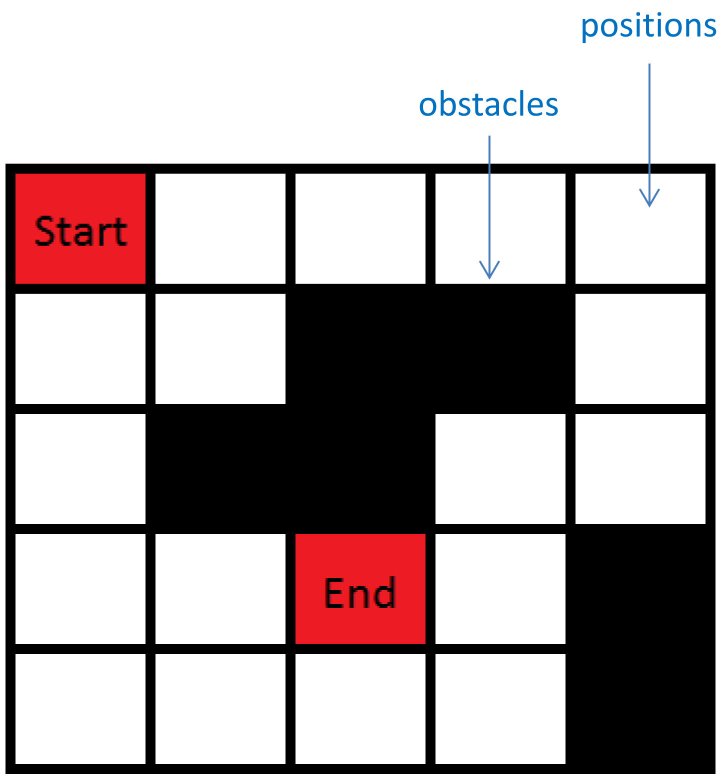
Path Planning

- There are 3 paths shown (blue and gray)
- They have different
  - Lengths
  - Time taken
  - Mode of transport
  - Price, etc.
- Length, time, etc. can be thought of as the ‘cost’ of the path
- In our case, we will consider length as the only ‘cost’ of the path



Paths for Robots

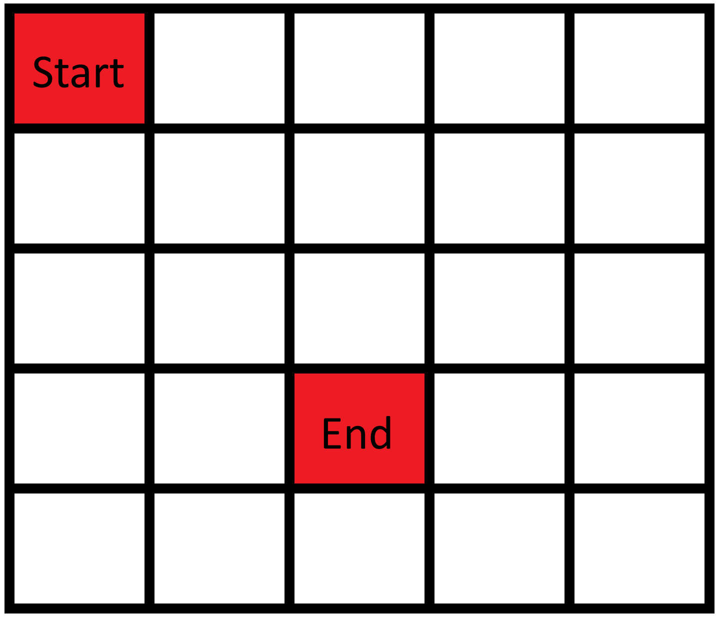
- We will simplify our problem in order to apply it for our robot path planning
- Paths are described by the positions that the robots pass through
- Regions where the robot cannot go are called **obstacles**
- The shortest path is one that passes through the least number of positions, while not passing through obstacles



Example of Path Search

Example:

- Given this grid, find a path between the Start and the End
- Rules:
  - The robot can only move horizontally or vertically



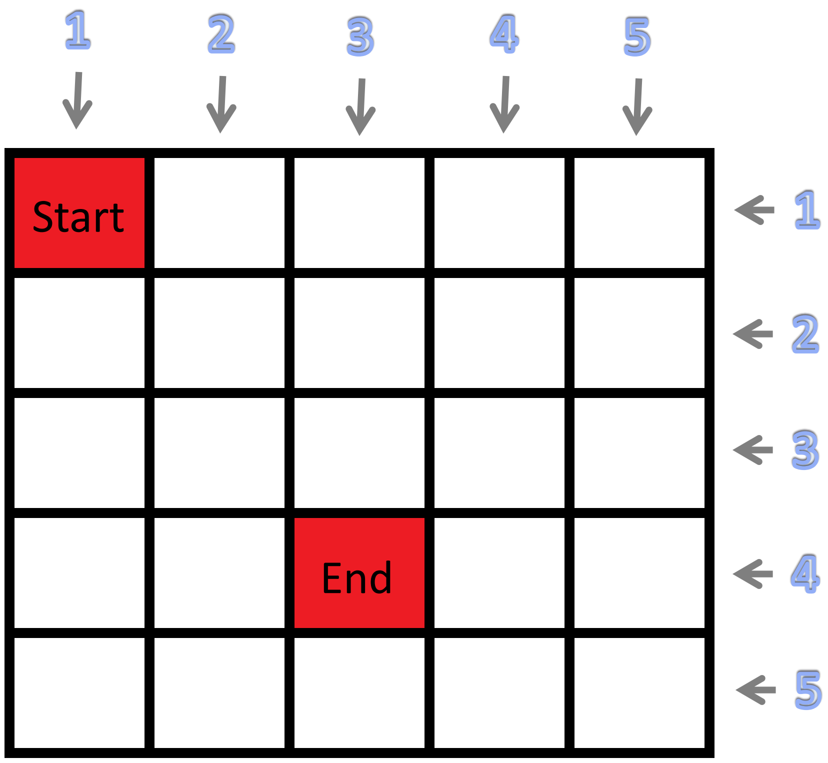
How do we start?

Since a path has a Start and an End, we need to first define these positions.

Example:

Start – (1,1)

End – (3,4)

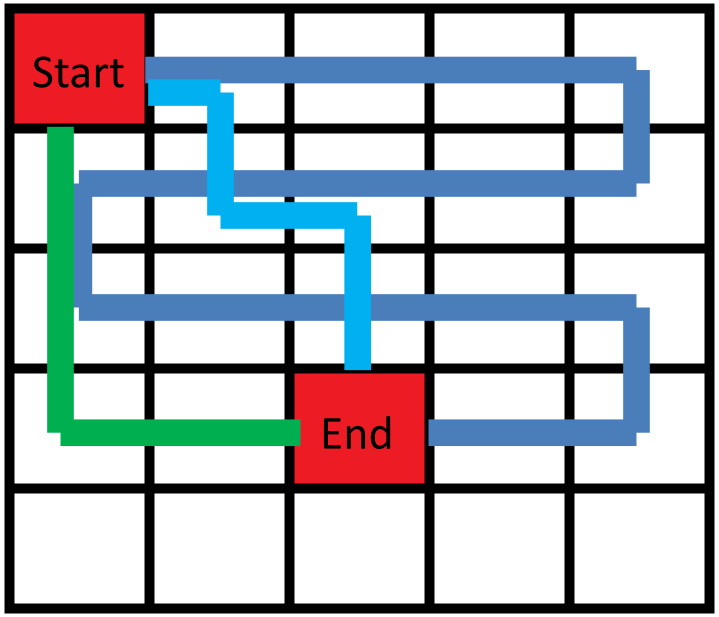


Path Search

Example:

- Possible solutions

Can we find an algorithm that will certainly give us the shortest path from Start to End?

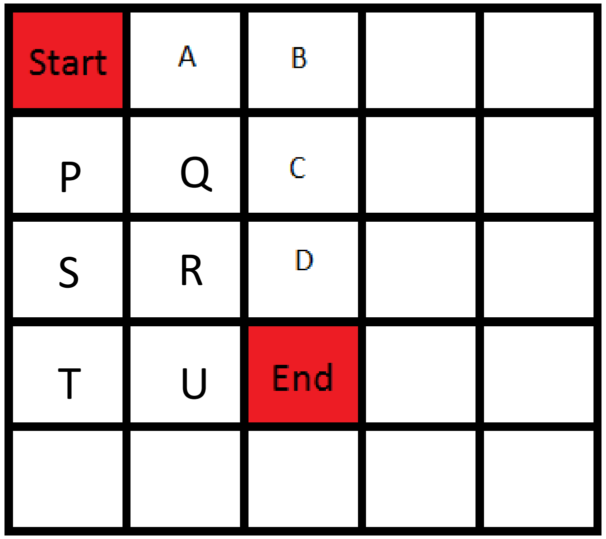


Path Search Algorithm

- One of the algorithms that can give us the shortest path is called the Dijkstra's Algorithm
- Assume that we know a shortest path,

  Start->P->Q->R->U->End
- Let us denote the length of the shortest path to a square N as len(N)
- Then we can say

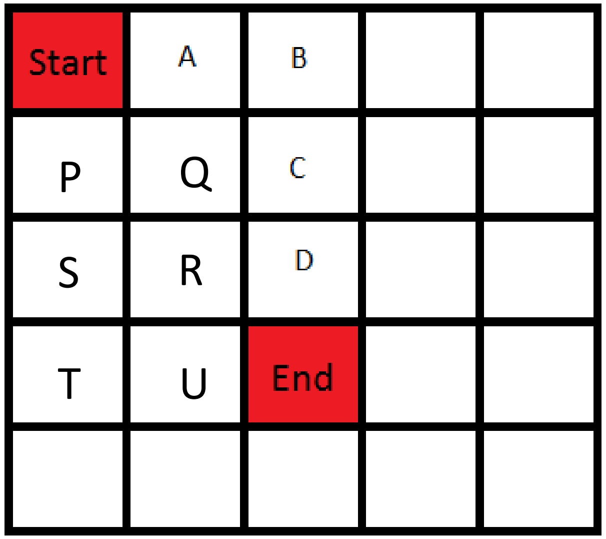
  len(End) = len(U) + 1
- Therefore, we must find the shortest path to U, that is, len(U) = len(R) + 1
- Therefore, we must find the shortest path to each square that may lie on the shortest path to the End



Robotics Competition Plus Pilot

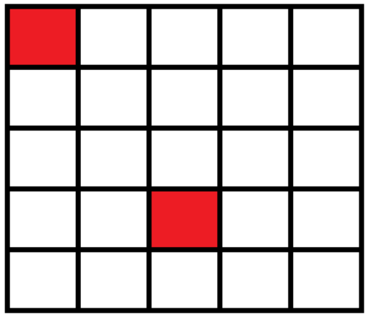
Path Search Algorithm

- So, for each square,
  - Find the shortest path
  - Find the square(parent) on that path which leads to the present square
- So we start by assigning the length of the shortest path to each square, starting from Start
- Step 0 :
  - len(Start) = 0, no parent square
- Step 1:
  - len(A) = 1, parent = Start
  - len(P) = 1, parent = Start
- Step 2:
  - len(B) = 2, parent = A
  - len (Q) = 2, parent = P (Note: we have not used A as a parent as we need only one shortest path)
  - len(S) = 2, Parent = P
- Step 3:
  - len(R) = 3, parent = Q
  - and so on…
- Step 5:
  - len(End) = 5, parent = U
- Once we have found len(End), we can not only conclude the length of the shortest path, but by tracing back the parents, we can specify exactly at least one shortest path that goes from the Start to the End

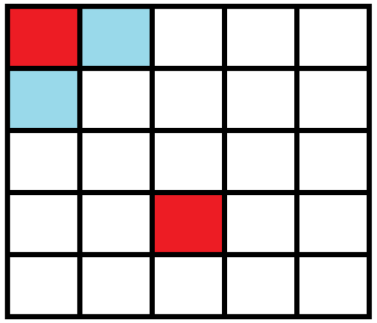


For a more comprehensive study, please look at http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

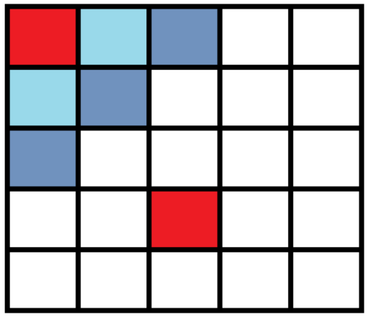
Start Searching



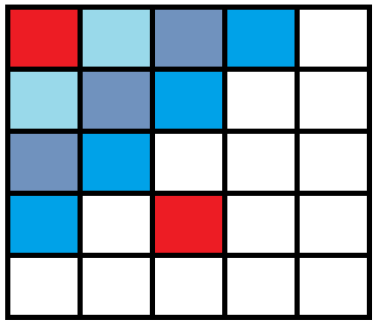
Step 0 –
len(Start) = 0
no parent square



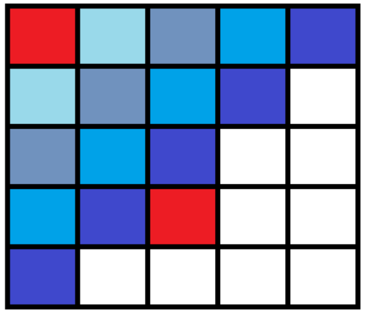
Step 1 –
len(A) = 1, parent = Start
len(P) = 1, parent = Start



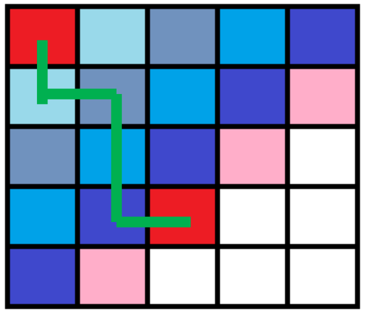
Step 2 –
len(B) = 2, parent = A
len (Q) = 2, parent = P
len(R) = 2, Parent = P



Step 3 …



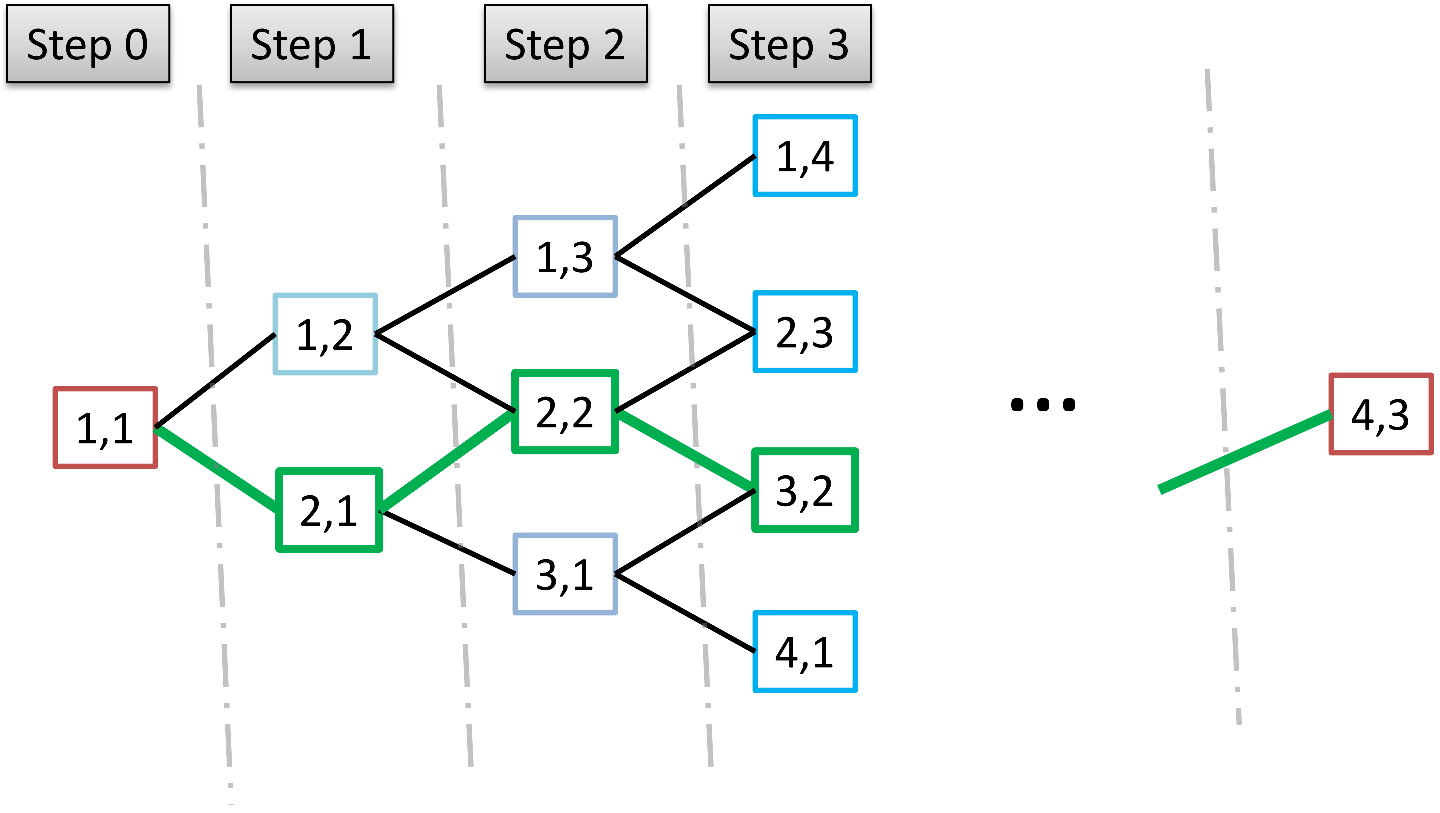
Step 4 …



Step 5 – End Found!

For animation, please look at https://www.youtube.com/watch?v=cSxnOm5aceA

In other words, trees!



The length of the path is the number of nodes excluding the start!