

①

Syed Muhammed Ghani

2020-CS-669

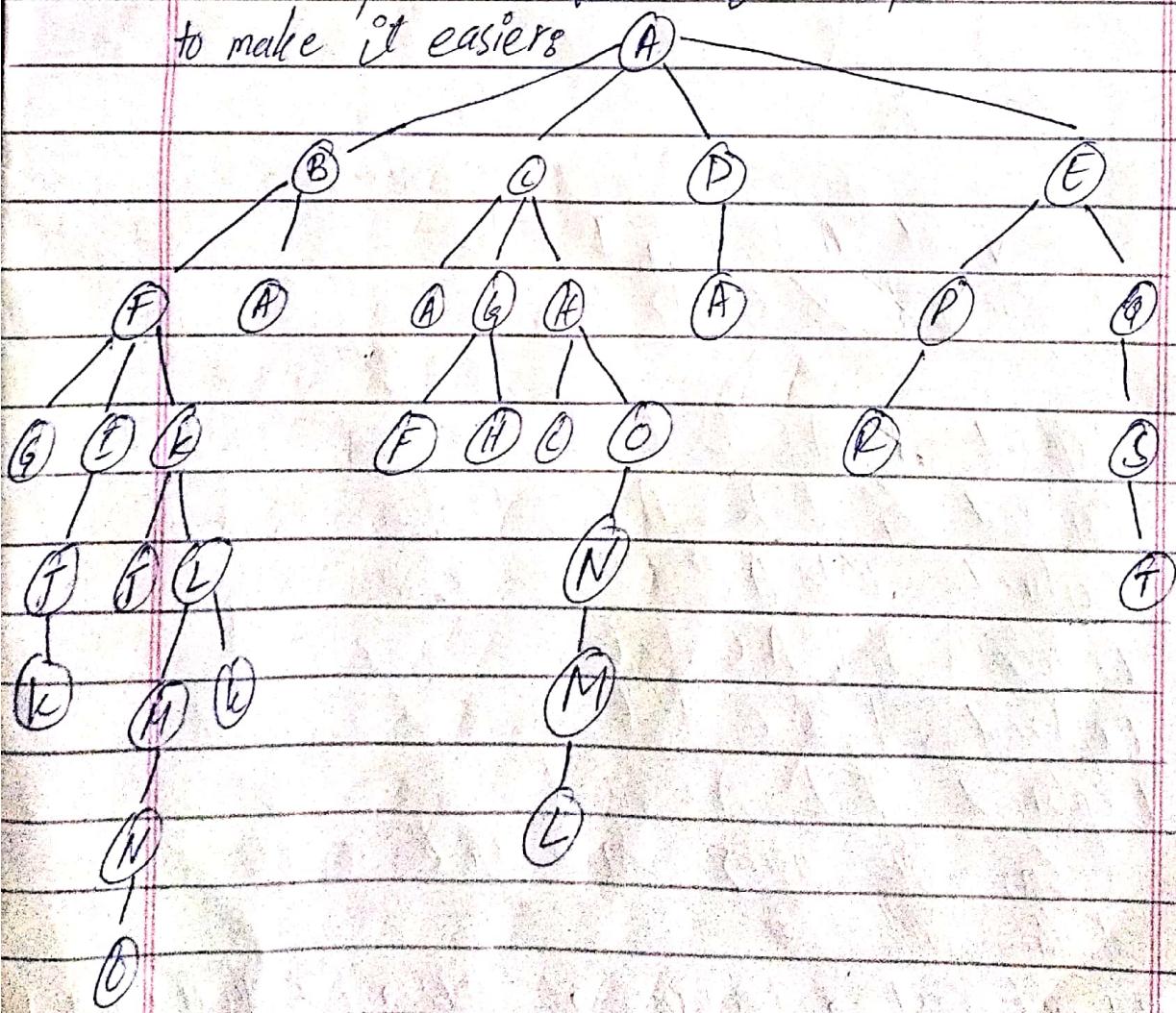
Artificial Intelligence

Assignment #1

Searching techniques:

Tree:

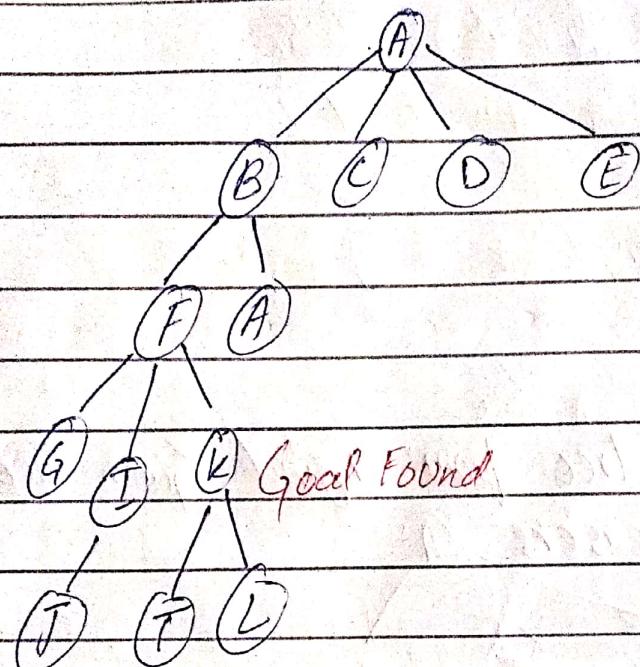
Sample tree from original given tree  
to make it easier



Q#18

(2)

① For Depth First search  
Starting Node A      Goal K



visited A B F G I J K

Explored A B C D E

Explored A C D E

So, by using Depth First search  
Path will be

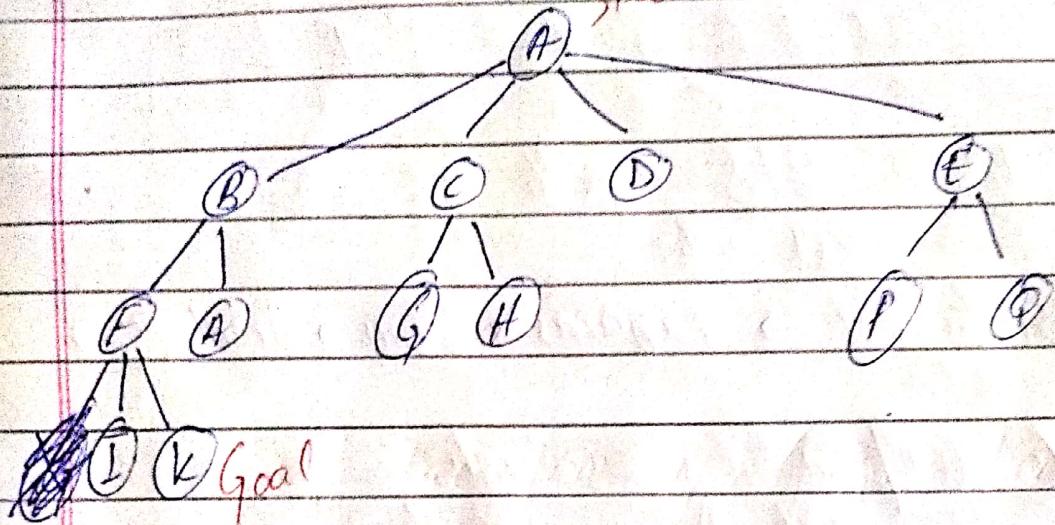
[start-Bucharest → Fagaras → Sibiu → Rimnicu Vilcea]  
Goal (Arad) ← Zerind ← Oradea ←

(b) Breadth First search

start A

Goal K

Stack



Visited ABCDEFGHIJKLMNOPK

Explored A

Explored BCD E F

Explored C D E F

Explored D E F G H

Explored E F G H

Explored F G H P Q

Explored G H P Q I K

Explored H P Q I K F

Explored P Q I K F C O

Explored Q I K F C O R

(v)

Explored K F C O R S

Explored K F C O R S T Goal Found.

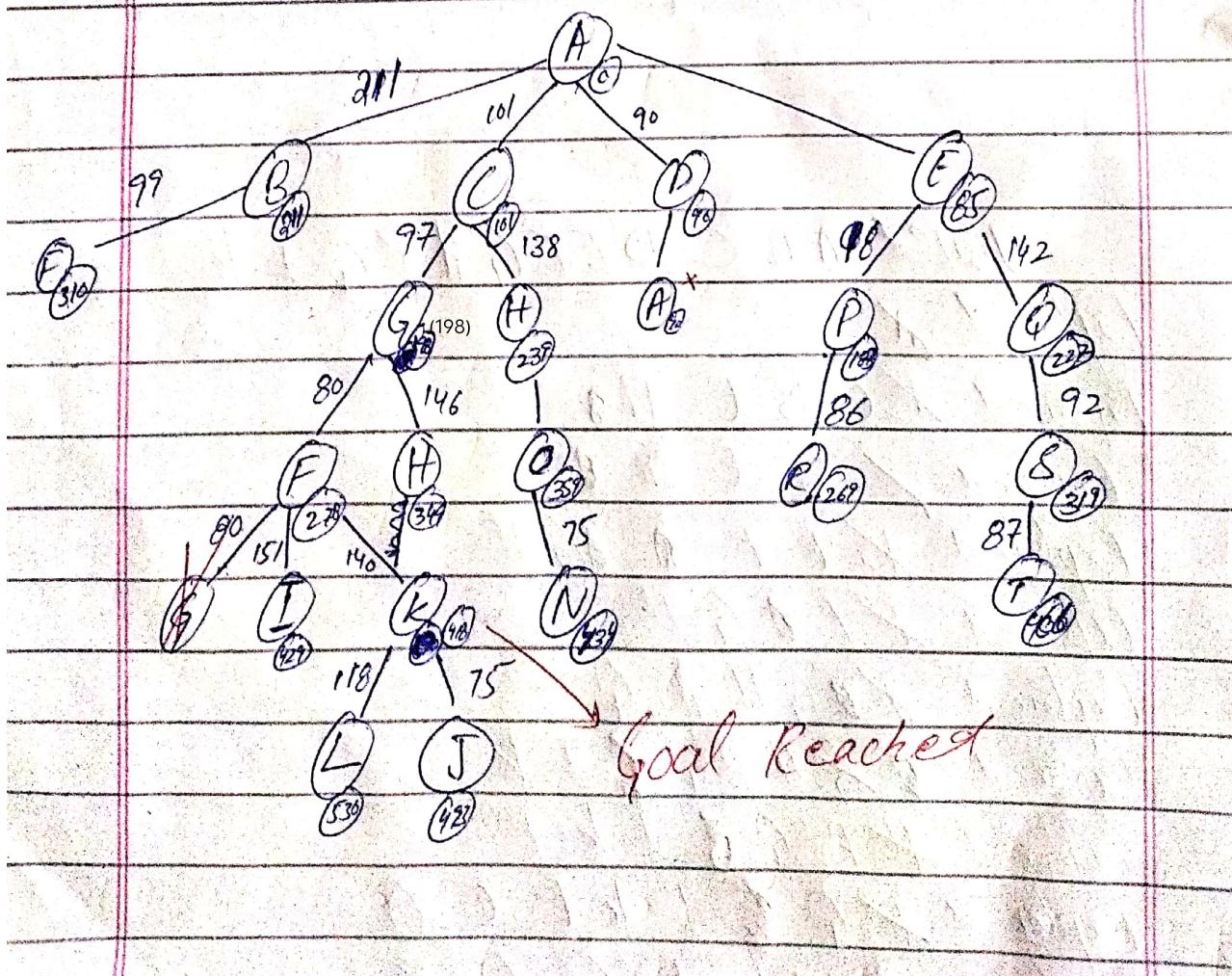
Explored F C O R S T L

Using BFS path is  
Shortest path

Start  $\rightarrow$  Bucharest  $\rightarrow$  Fagaras  $\rightarrow$  Sibiu  $\rightarrow$  Arad (Goal)

(c) Uniform Cost Search

Start: A      Goal: K



Goal Reached

(5)

Visited A E D C P G B Q H R F S O T K

Explored A<sup>0</sup>

Explored E<sup>65</sup> D<sup>90</sup> C<sup>101</sup> B<sup>211</sup>

Explored D<sup>90</sup> C<sup>101</sup> P<sup>183</sup> B<sup>211</sup> Q<sup>227</sup>

Explored E<sup>101</sup> P<sup>183</sup> B<sup>211</sup> Q<sup>227</sup>

Explored P<sup>183</sup> G<sup>198</sup> B<sup>211</sup> Q<sup>227</sup> H<sup>239</sup>

Explored G<sup>198</sup> B<sup>211</sup> Q<sup>227</sup> H<sup>239</sup> R<sup>269</sup>

Explored B<sup>211</sup> Q<sup>227</sup> H<sup>239</sup> R<sup>269</sup> F<sup>278</sup>

Explored Q<sup>227</sup> H<sup>239</sup> R<sup>269</sup> F<sup>278</sup>

Explored H<sup>239</sup> R<sup>269</sup> AF<sup>278</sup> S<sup>319</sup>

Explored R<sup>269</sup> F<sup>278</sup> S<sup>319</sup> O<sup>359</sup>

Explored F<sup>278</sup> S<sup>319</sup> O<sup>359</sup>

Explored S<sup>319</sup> O<sup>359</sup> K<sup>418</sup> I<sup>429</sup>

Explored O<sup>359</sup> K<sup>418</sup> I<sup>429</sup> L<sup>429</sup> J<sup>429</sup>

Explored J<sup>429</sup> K<sup>418</sup> I<sup>429</sup> N<sup>434</sup>

Explored L<sup>429</sup> I<sup>429</sup> N<sup>434</sup>

Goal Found

So, using Uniform Cost Search

- Minimum cost is 418

Path is: Minimum

A → C → G → F → k

For actual Graph:

Bucharest → Pitesti → Rimnicu Vilcea  
Arad ← Sibiu

# (6)

## Iterative Deepening Depth Limited Search:

- Iterative Depth search with threshold 3:

1st iteration:  $d=0$

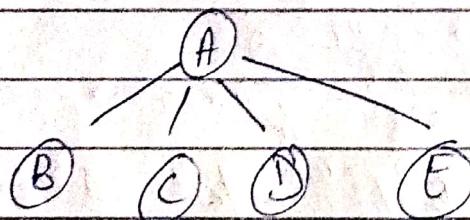


visited A

Goal Not Found

Explored A B C D E

2nd iteration:  $d=1$



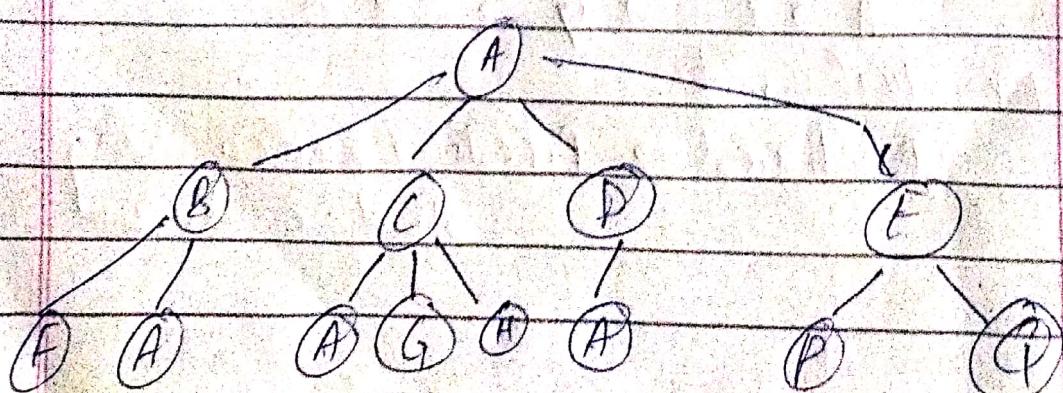
visited A

visited A B C D E

Explored A B C D E

Goal Not Found

3rd iteration:  $d=2$



Visited A B F C G H D E P Q

Explored A B C D E

Explored F C D E

Explored F D E

Explored G H C D E

Explored H C D F

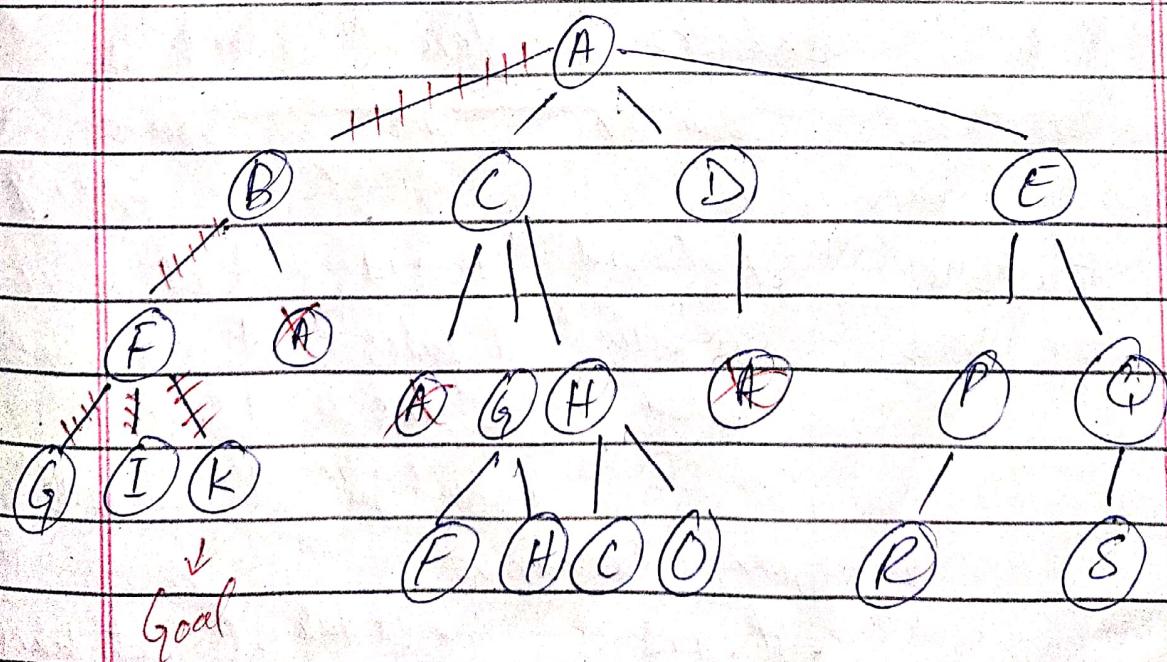
Explored C D E

Explored B E

Explored P Q

Explored Q

4th Iteration  $d=3$



(8)

Visited A B F G I k

Explored A B C D E

Explored F C D E

Explored G I k C D E

Explored I k C D E

Explored K C D E → Goal found

Explored C D E

Using Iterative deepening search

Goal found in 4th iteration  
with path

Start → Bucharest → Fagaras → Sibiu  
(Goal) Arad ← Oradea ← Rimnicu Vilcea ←

#### (d) Comparison Between different searches

	Completeness	Optimality	Time	Space
	Complexity			Complexity
DFS	it ensure completeness	it doesn't give optimal solution	In worst case it takes b <sup>d</sup> time to find Goal	it occupy more space in worst case is b <sup>d</sup> .

In Given Scenario:

• it finds Goal	• it give optimal solution as compared to BFS	• It takes less time BFS	it takes less Space than BFS
-----------------	---	--------------------------	------------------------------

## BFS

	completeness	optimality	time complexity	space complexity
--	--------------	------------	-----------------	------------------

BFS	Yes	No	$b^d$	$b \times d$
-----	-----	----	-------	--------------

In Given problems

	Yes	No	More time than DFS	More space than DFS
--	-----	----	--------------------	---------------------

## U-COS

	optimality	completeness	time complexity	space complexity
U-COS	Yes	Yes	$b^{1 + C/\epsilon}$	$b^{1 + C/\epsilon}$

in Give case

- This technique Gives optimal solution in case of cost function.
- it provide Best path with Minimum cost.

## IDS

	optimal	complete	time	space.
--	---------	----------	------	--------

IDS	More optimal than BFS	Ensure solution	less time but	less space than DFS
-----	-----------------------	-----------------	---------------	---------------------

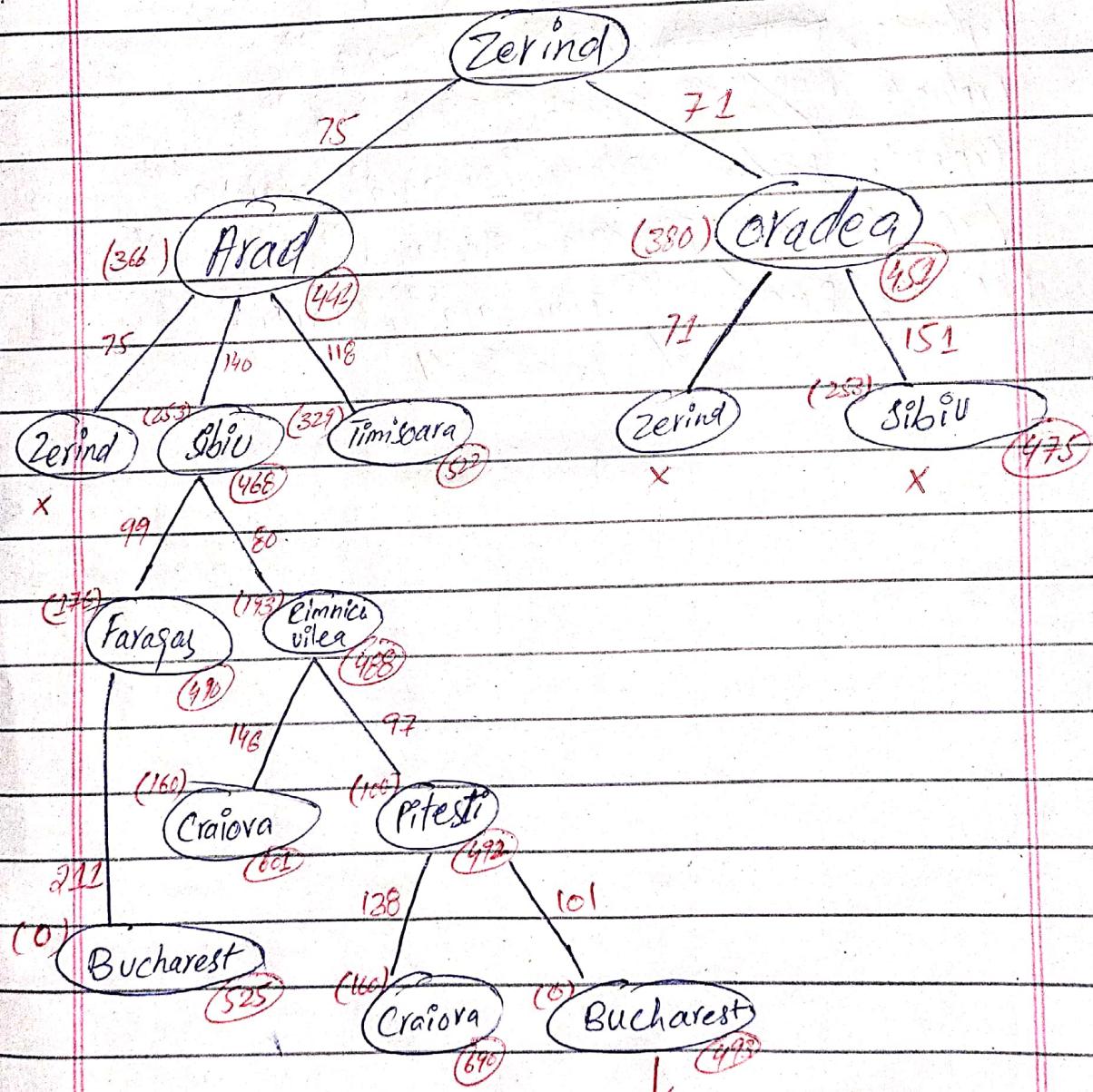
SDFS	More iterations	than BPS
------	-----------------	----------

G#2

(a) A\* Search

Start: Zerind

Goal: Bucharest



Visited: Zerind | Arad | Oradea | Sibiu | Rimnicu Vilcea | Pitesti | Bucharest

Explored: Zerind

Explored: Arad <sup>441</sup> Oradea <sup>451</sup>

Explored: Oradea <sup>451</sup> Sibiu <sup>468</sup> Timisoara <sup>522</sup>

Explored: Rimnicu Vilcea <sup>488</sup> Sibiu <sup>468</sup> Timisoara <sup>522</sup>

Explored: Rimnicu Vilcea <sup>488</sup> Faragash <sup>490</sup> Timisoara <sup>522</sup>

Explored: Faragash <sup>490</sup> Pitesti <sup>492</sup> Timisoara <sup>522</sup> Craiova <sup>601</sup>

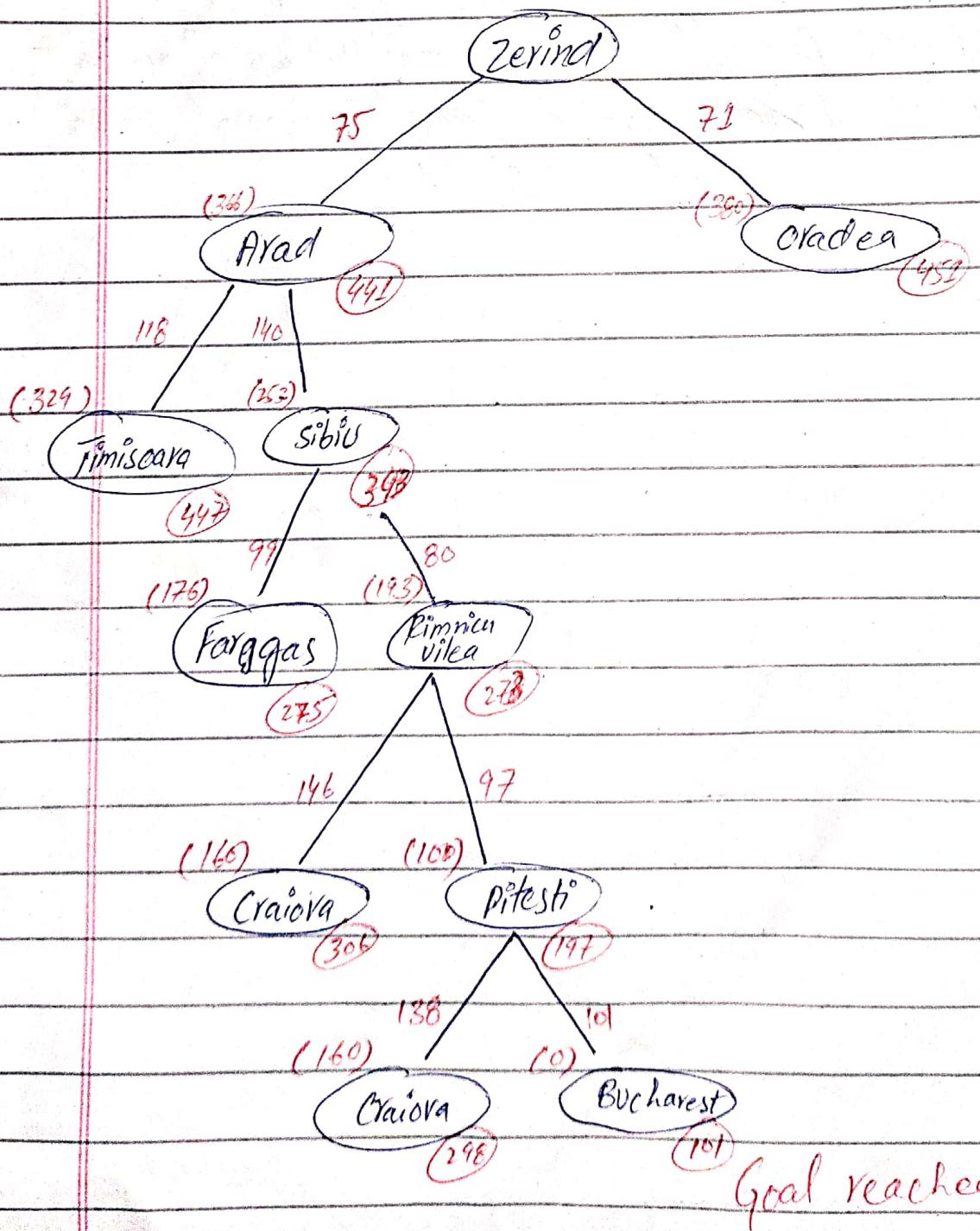
Explored: Pitesti <sup>492</sup> Timisoara <sup>522</sup> Bucharest <sup>525</sup> Craiova <sup>601</sup>

Explored: Bucharest <sup>493</sup> Timisoara <sup>522</sup> Craiova <sup>601</sup>

Goal Reached!

## (b) Recursive Best First Search:

Start Zerind      Goal Bucharest.



Visited: Zerind Arad Sibiu

Explored: Zerind

Explored: Arad 441 Oradea 451

Explored: Sibiu 393 Timisoara 447 Oradea 451

Explored: Rimnicu Vilcea 273 Faragash 275 Timisoara 447 Oradea 451

Explored: Petesti 197 Craiova 306 Faragash 275 Timisoara 447 Oradea 451

Explored: Bucharest 101 Craiova 298 Faragash 275 Timisoara 447 Oradea 451

Goal Reached!

### (C) Comparison between A\* search vs RBFS

	Completeness	Optimality	Time complexity	Space complexity
A*	Yes	Yes	$b^d$	$b^d$
RBFS	Yes	Yes	$b^d$	$b^d$
	Better than	if heuristic is admissible		

Overall RBFS is much better than A\* as RBFS visits lesser nodes than A\* so, it give optimized solution.