

## Heuristic / Informed Search

— Heuristics are:

— criteria

— Methods

— Principles

• for deciding which among the several alternatives of actions promises to be the most effective in order to achieve goal.

## Heuristic function: $h(n)$

\* A heuristic function of node ' $n$ ' is an estimate of the optimum cost from the current node to a goal node.

\* It is denoted by  $h(n)$

\*  $h(n)$  = estimated cost of the cheapest path from node ' $n$ ' to goal node.

For example!

8-puzzle : Misplaced Tiles Heuristics is the number of tiles out of place.

2	8	3
1	6	4
	7	5

Initial state

fig (a)

1	2	3
8		4
7	6	5

Goal state

fig (b)

\* Here, fig (a) shows the current state

n, & the fig (b) present the goal

state. So,  $h(n) = 5$

\* Because tiles 2, 8, 1, 6, & 7 are out of

place.

## Best First Search:

- evaluation function:  $f(n)$ 
  - gives an estimate
  - node with lowest cost is expanded first.
- Key component of  $f(n)$  is  $h(n)$ , which is additional knowledge of the problem.

\* Based on evaluation function

$f(n)$

— Greedy Best-first search

— A\* search

### Greedy Best-first Search

— It uses evaluation function  $f(n)$  to select which node is to be expanded.

— Node with lowest evaluation

cost is selected for expansion.

Evaluation function:

$$f(n) = h(n) \text{ (heuristic)}$$

i.e. estimate of cost from  
node  $n$  to goal node.

For example:

$h_{SLD}(n)$  = straight line distance  
from node  $n$  to goal

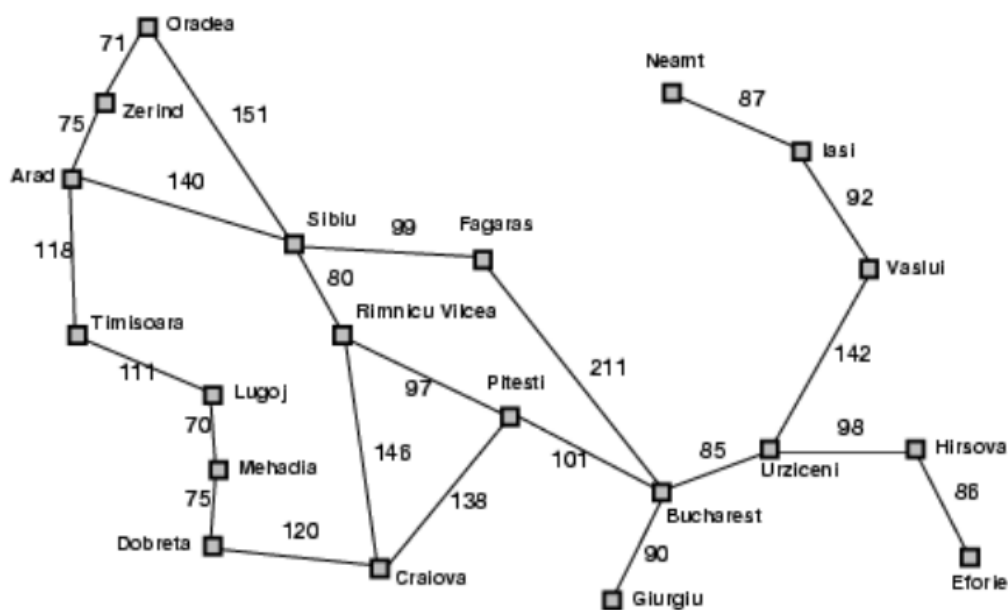
- greedy best-first search expand  
the node to be appear closest  
to the goal.

✓  
- Time & space complexity is:

$$O(b^m)$$

where,  $b$  is branching factor  
 $m$  is max depth

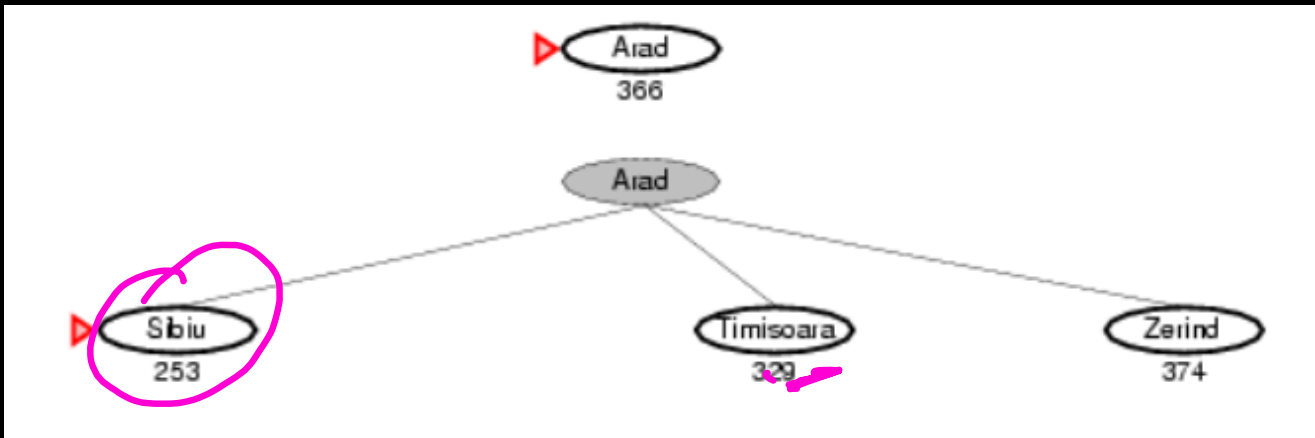
Example: Given following graph of cities  
starting at Arad, problem is to reach the  
Bucharest.



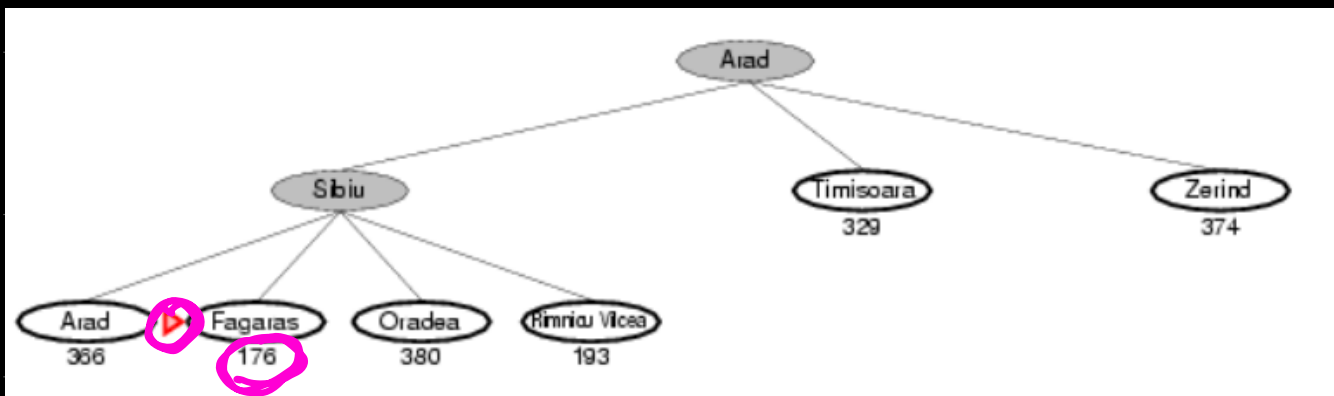
Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Step 1:

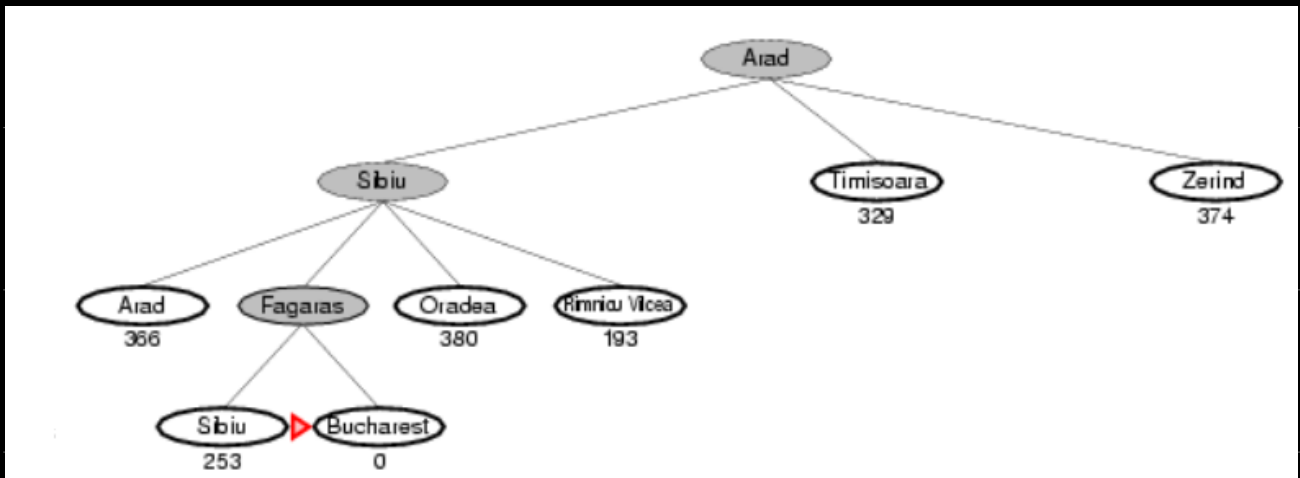


Step 2:



Step 3:





## A\* Search

– it is best-first informed graph search algorithm.

– It uses the heuristic  $h(n)$  & path cost  $g(n)$

i.e. evaluation function

$$f(n) = h(n) + g(n)$$

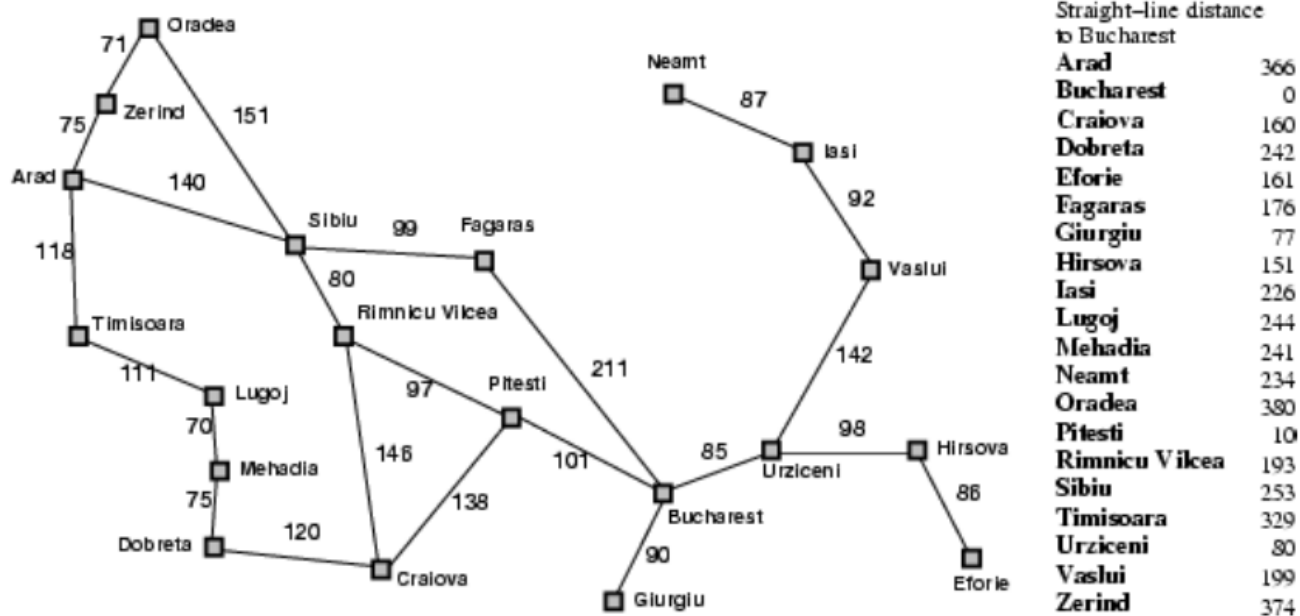
where,

$f(n)$  is estimated total cost of path  
through  $n$  to goal

$h(n)$  is estimated cost to goal  
from  $n$

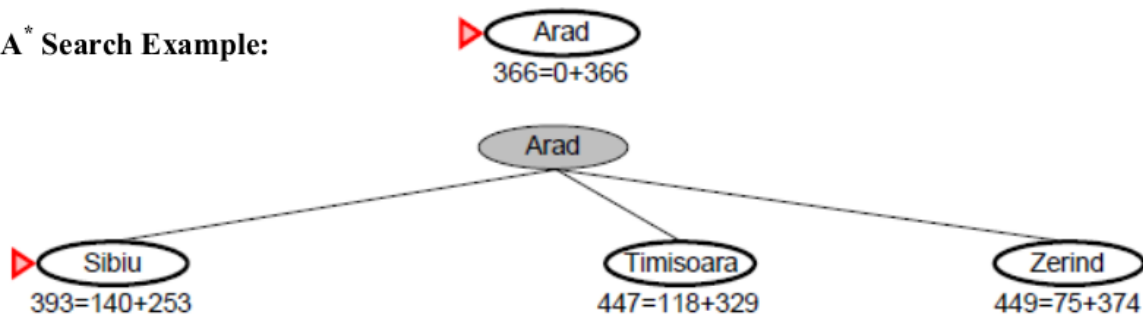
$g(n)$  cost so far to reach  $n$   
from root.

Example:  $A^*$  Search

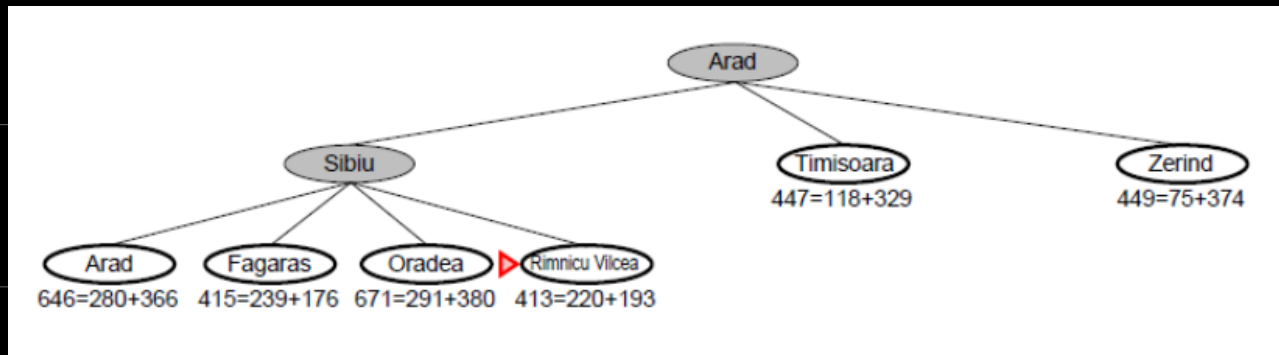


Step 1:

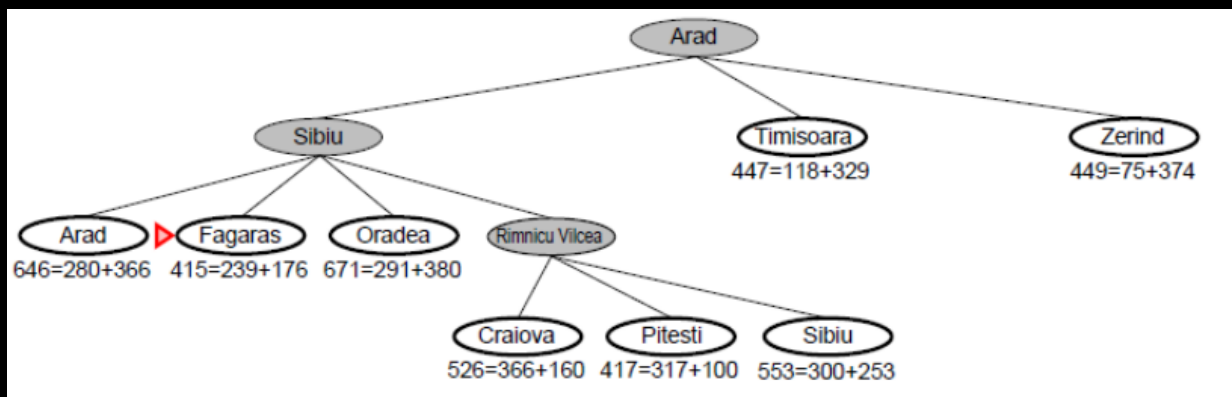
A\* Search Example:



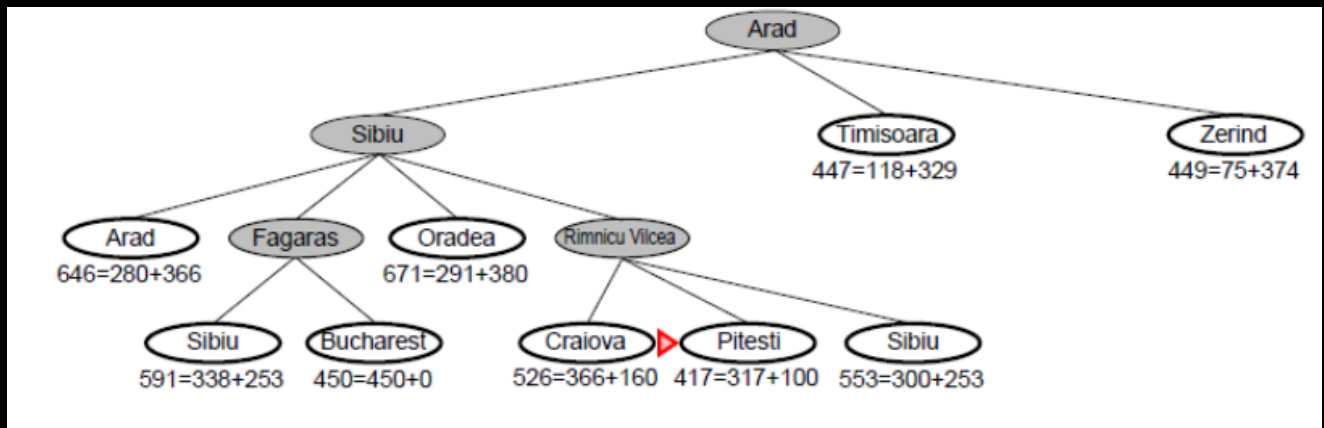
Step 2



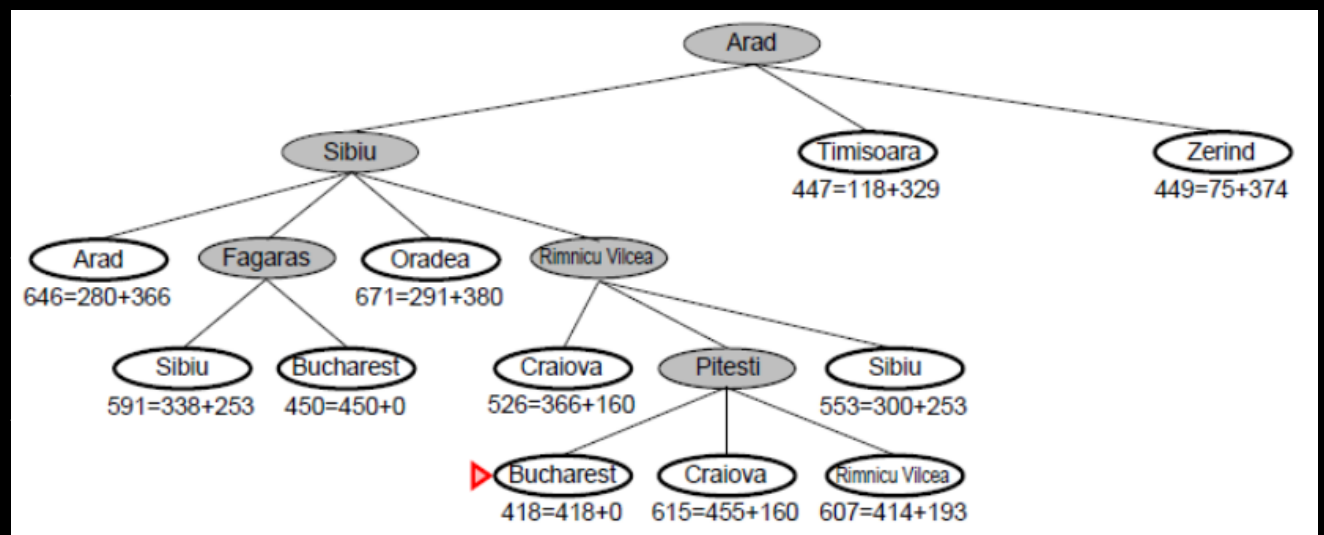
Step 3



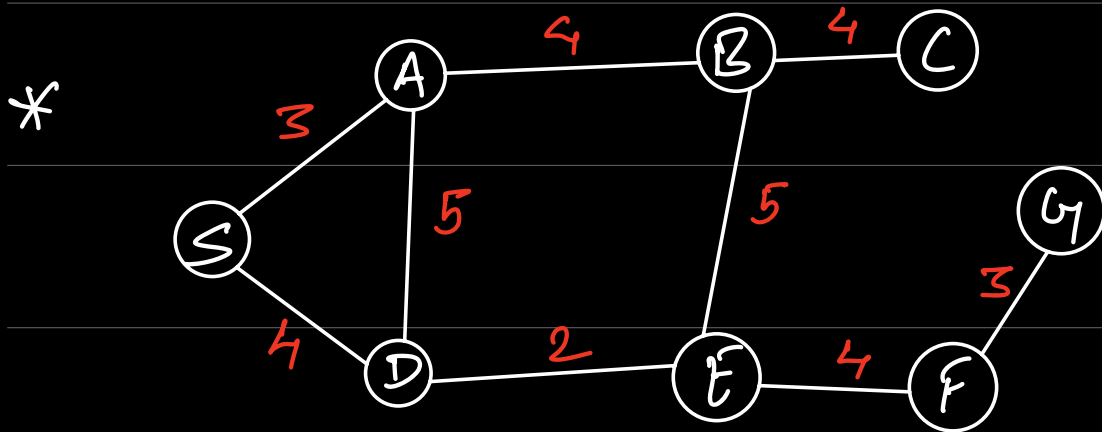
Step 4



Step 5

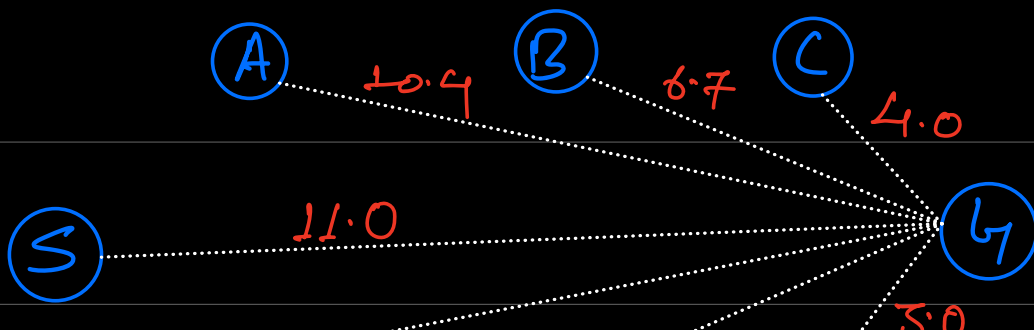


Solve the following searching problem using greedy best-first &  $A^*$  searching techniques.



Here, S is starting node & G is the goal node.

The straight line distance is given as



ⓓ

8.9

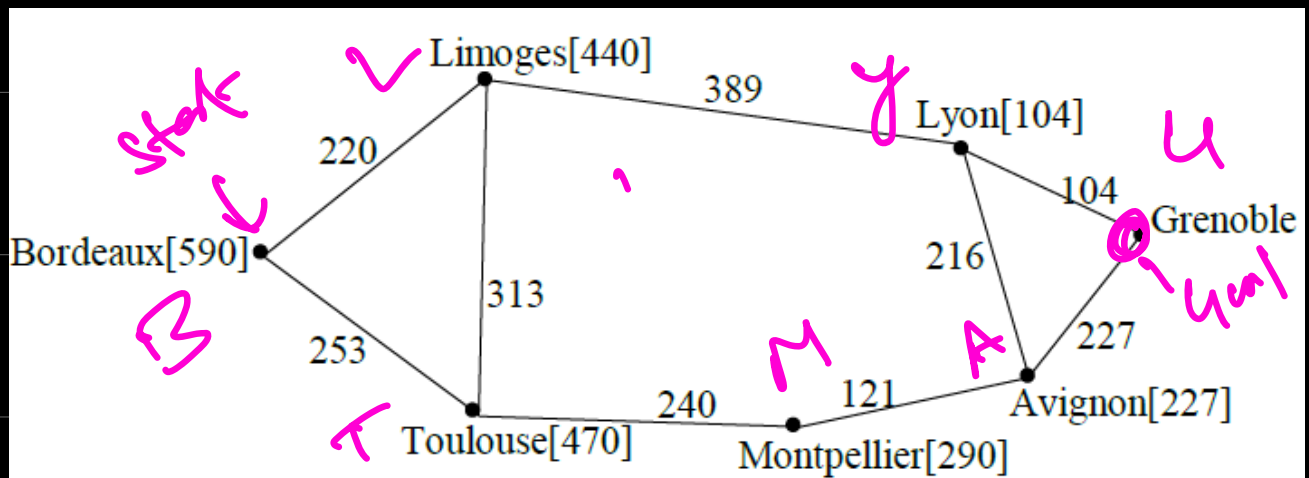
ⓔ

6.9

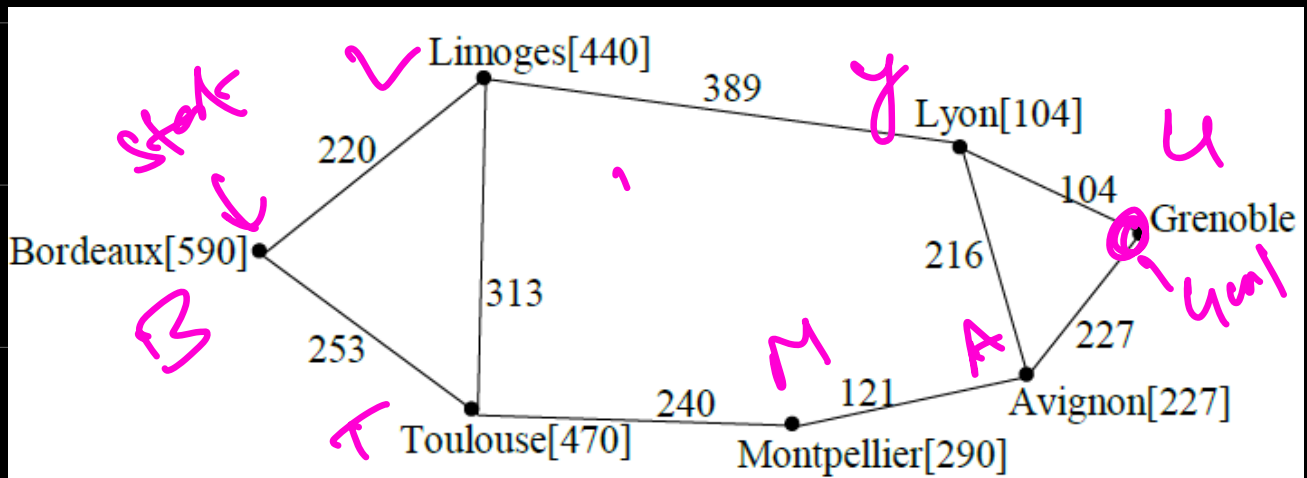
ⓕ

...

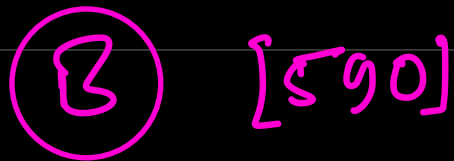
\* Consider the following map of French cities



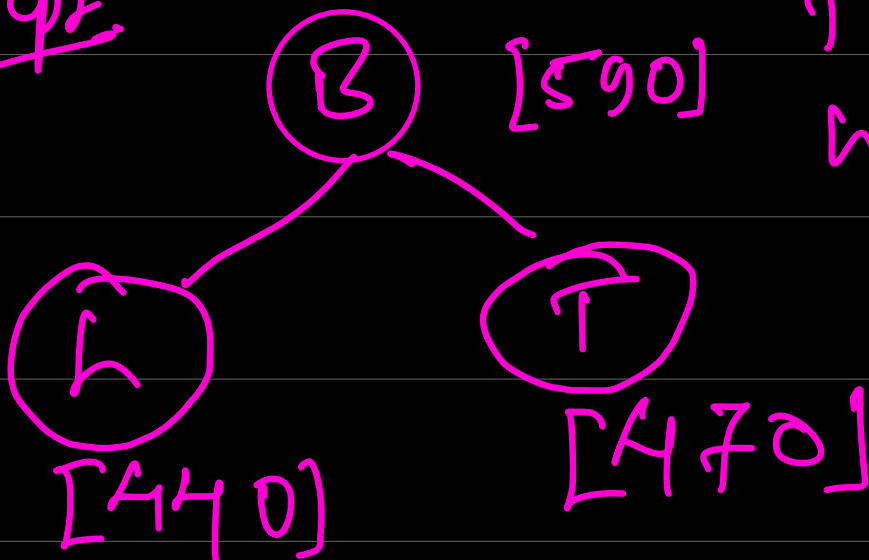
Consider,  
start node is Bordeaux  
to goal node to be  
Grenoble.



Step 1:



Step 2:

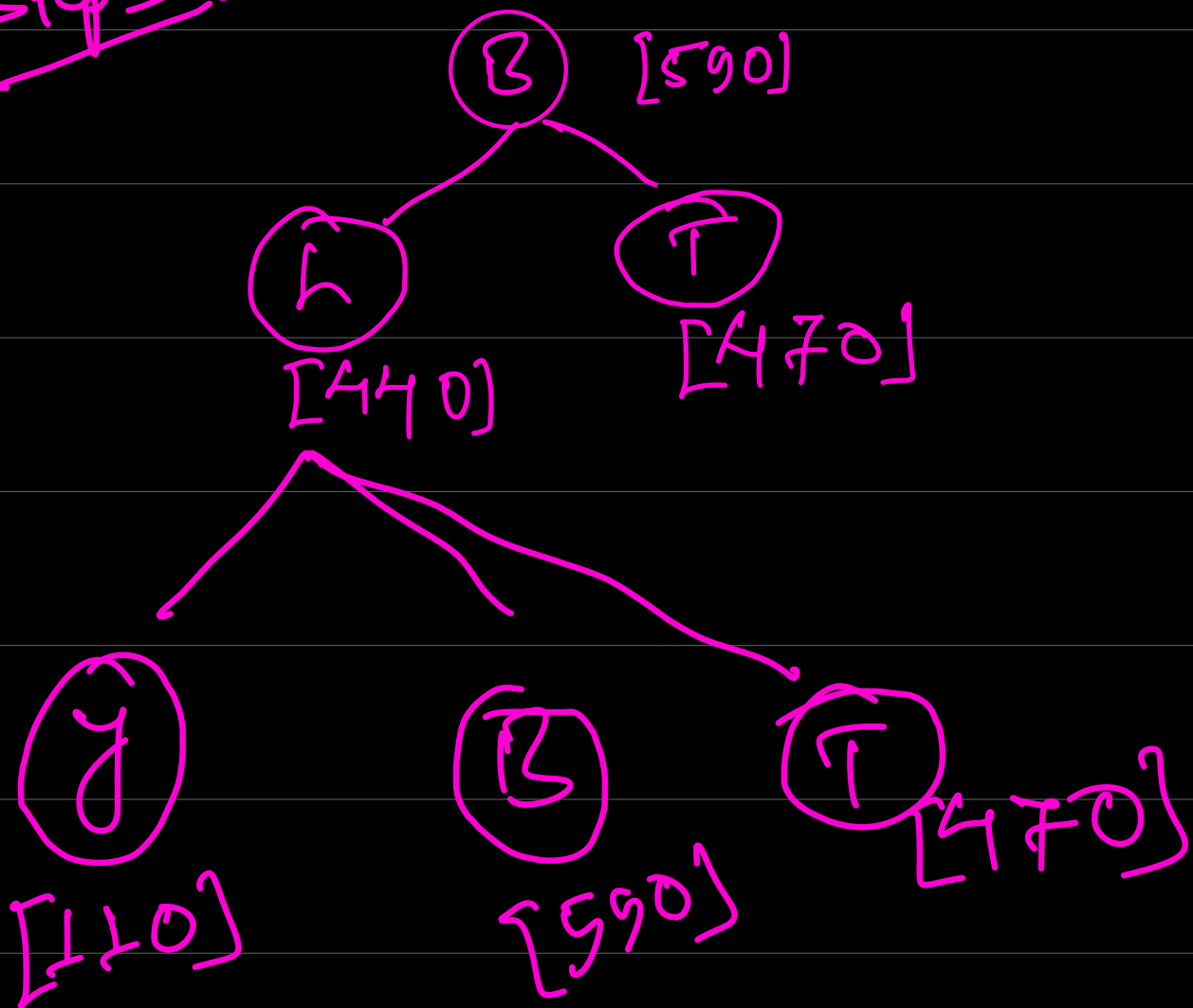


$$f(u) = h(u)$$

$$h(u) = \sum LD$$

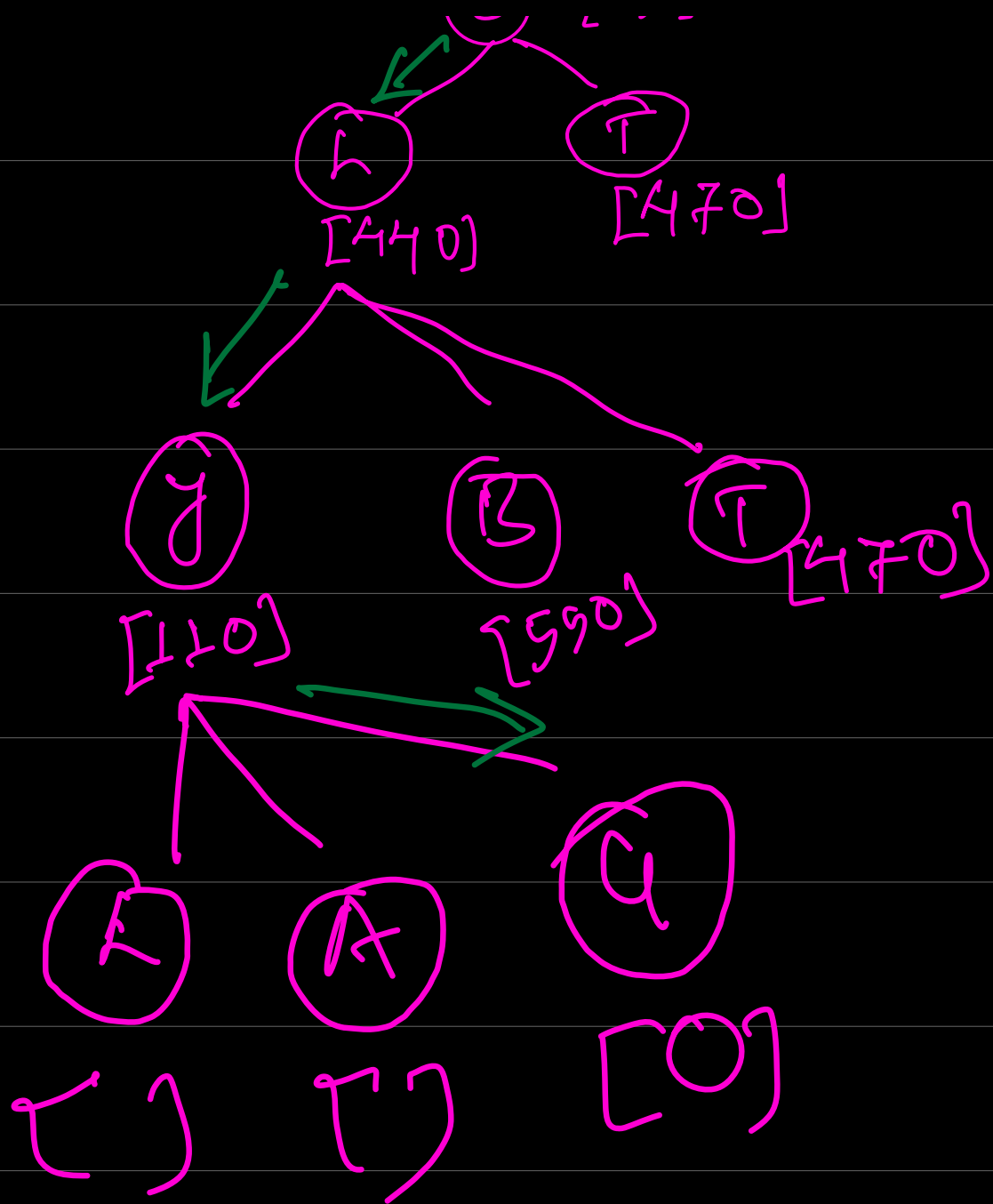


step 3:



step 4:

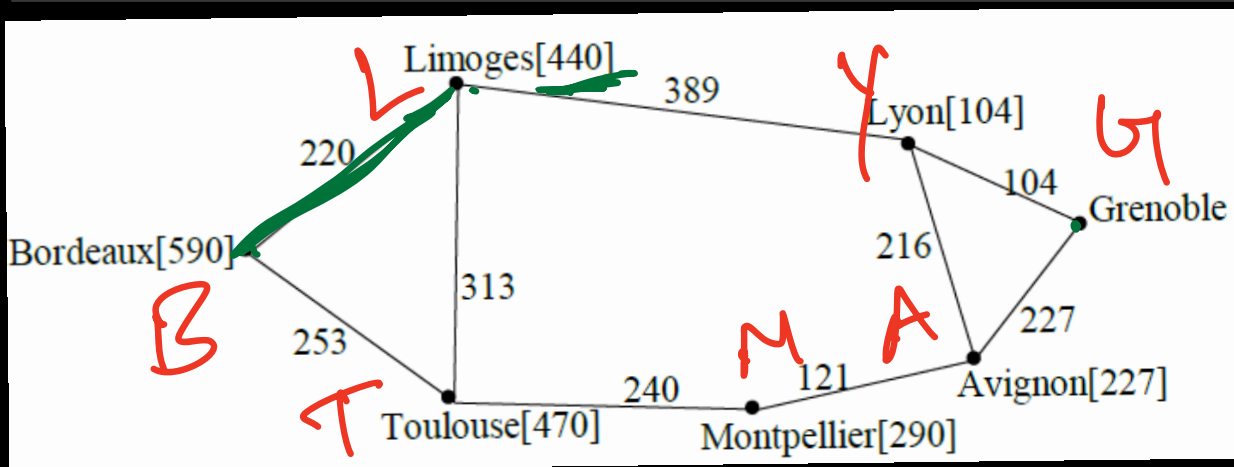
(B) [590]



$f(n) = h(n)$   
 ↳ straight  
 line distance  
 n to

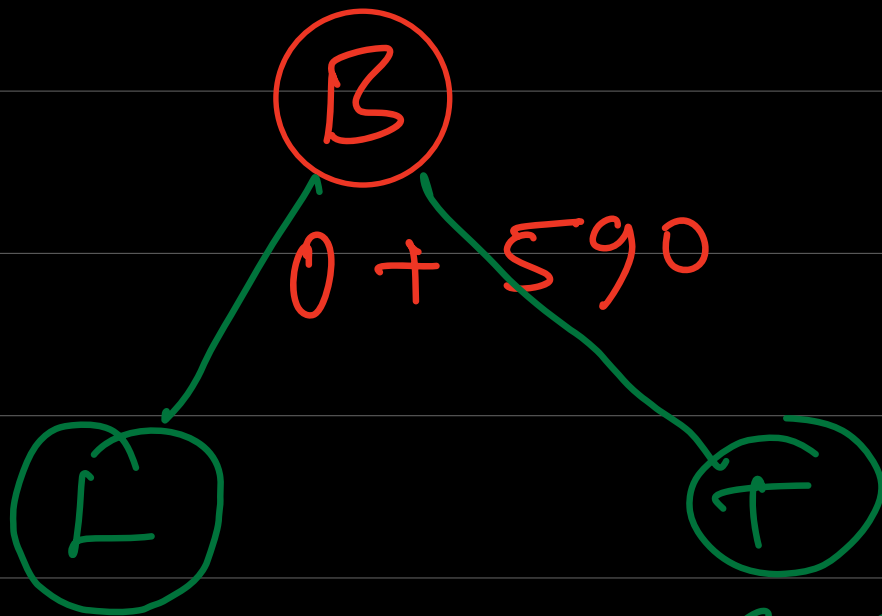
from rose  
year 4.

A\*



evaluation func:

$$f(n) = (g(n) + h(n))$$



$$\underline{220} + 440 = 660 \quad 253 + 470 = 723$$