

# Assignment 2 Report

Option chosen is:

**b. Show Breadth First Search and A\* search on this data.**

## 1. Breadth-First Search

Here I have implemented the bfs algorithm to calculate and print the distance between source city and destination city. It traverses through the branches and neighbours accordingly and gives the result as distance in km.

```
?- start.

Welcome to the Road Distance Caculator!
List of Cities:
Agartala Agra Ahmedabad Allahabad Amritsar Asansol Bangalore B
aroda Bhopal Bhubaneshwar Bombay Calcutta Calicut Chandigarh C
ochin Coimbatore Delhi Gwalior Hubli Hyderabad Imphal Indore
Jabalpur Jaipur Jamshedpur Jullundur Kanpur Kolhapur Lucknow L
udhiana Madras Madurai Meerut Nagpur Nasik Panjim Patna Pondi
cherry Pune Ranchi Shillong Shimla Surat Trivandrum Varanasi
Vijayawada Vishakapatnam
Calculate distance
From 'Agra'.

To |: 'Agartala'.

How do you want to calculate distance:
1. BFS
2. A* Search
|: 1.

Path is found, calculating ...

Route : [Agra,Ahmedabad,Agartala]
Distance : 4183
true .

?- ■
```

## 2. A\* search

Here I have implemented the A\* search algorithm to calculate and print the distance between source city and destination city. It follows  $f(n) = g(n) + h(n)$  to do so, where  $g(n)$  is the distance till that city,  $h(n)$  is the heuristics.

```
?- start.

Welcome to the Road Distance Calculator!

List of Cities:
Ahmedabad, Bangalore, Bhubaneshwar, Bombay, Calcutta, Chandigarh, Cochin, Delhi, Hyderabad, Indore, Jaipur,
Jin, Patna, Pondicherry, Pune

Calculate distance
From 'Vishakapatnam'.
To |: 'Vijayavada'.

How do you want to calculate distance:
1. BFS
2. A* Search
|: 2.

Path is found, calculating ...

currently on Vishakapatnam
Vijayavada
[Vishakapatnam, Bhubaneshwar]
currently on Bhubaneshwar
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur]
currently on Jamshedpur
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta]
currently on Calcutta
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol]
currently on Asansol
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna]
currently on Patna
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi]
currently on Varanasi
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow]
currently on Lucknow
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur]
currently on Kanpur

SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur]
currently on Jamshedpur
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta]
currently on Calcutta
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol]
currently on Asansol
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna]
currently on Patna
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi]
currently on Varanasi
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow]
currently on Lucknow
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur]
currently on Kanpur
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur, Allahabad]
currently on Allahabad
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur, Allahabad, Nagpur]
currently on Nagpur
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur, Allahabad, Nagpur, Jabalpur]
currently on Jabalpur
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur, Allahabad, Nagpur, Jabalpur, Hyderabad]
currently on Hyderabad
Vijayavada
[Vishakapatnam, Bhubaneshwar, Jamshedpur, Calcutta, Asansol, Patna, Varanasi, Lucknow, Kanpur, Allahabad, Nagpur, Jabalpur, Hyderabad, Vijayavada]
ending
Route : _30
Distance : _32
true .

?-
```

### 3. Features used:

```
findall(  
    Distance_fn - X,  
    (  
  
        get_heuristics(From_city , X , Distance_between_cities ,  
Node_Heuristics),  
        From_city \== X,  
        not(member(X , Psf)),  
  
        Distance_fn is Distance_between_cities + Node_Heuristics  
  
    ),  
    PairList  
) ,
```

```
sort(PairList , Sorted_List) ,  
  
get_head(Sorted_List , Head) ,  
  
remove_head(Sorted_List , Returned_List) , ! ,  
  
pairs_keys_values([Head] , [Key] , [Value]) ,  
  
append(Psf , [Value] , New_psf) ,  
  
not(member(Value , Psf)) , ! ,  
  
write(New_psf) , nl ,
```

Lists, Write/ Read, Recursion, Backtracking, Pair, Key-Value list.  
Library functions append, pairs, member etc,

### 4. Heuristics:

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project named 'WALLPAPS'. The file 'heuristics.pl' is selected. The main editor shows the following Prolog code:

```
1 heuristics('Agartala', 'Ahmedabad', 3305, 3205).
2 heuristics('Agartala', 'Bangalore', 3824, 3724).
3 heuristics('Agartala', 'Bhubaneswar', 2286, 2186).
4 heuristics('Agartala', 'Bombay', 3593, 3493).
5 heuristics('Agartala', 'Calcutta', 1863, 1763).
6 heuristics('Agartala', 'Chandigarh', 2998, 2898).
7 heuristics('Agartala', 'Cochin', 4384, 4284).
8 heuristics('Agartala', 'Delhi', 2708, 2608).
9 heuristics('Agartala', 'Hyderabad', 3330, 3230).
10 heuristics('Agartala', 'Indore', 2891, 2791).
11 heuristics('Agartala', 'Jaipur', 2881, 2781).
12 heuristics('Agartala', 'Kanpur', 2281, 2181).
13 heuristics('Agartala', 'Lucknow', 2252, 2152).
14 heuristics('Agartala', 'Madras', 3493, 3393).
15 heuristics('Agartala', 'Nagpur', 2696, 2596).
16 heuristics('Agartala', 'Nasik', 3365, 3265).
17 heuristics('Agartala', 'Panjim', 3507, 3407).
18 heuristics('Agartala', 'Patna', 1681, 1581).
19 heuristics('Agartala', 'Pondicherry', 3661, 3561).
20 heuristics('Agartala', 'Pune', 3442, 3342).
21 heuristics('Agra', 'Ahmedabad', 878, 778).
22 heuristics('Agra', 'Bangalore', 1848, 1748).
23 heuristics('Agra', 'Bhubaneswar', 1578, 1478).
24 heuristics('Agra', 'Bombay', 1202, 1102).
25 heuristics('Agra', 'Calcutta', 1300, 1200).
26 heuristics('Agra', 'Chandigarh', 448, 348).
27 heuristics('Agra', 'Cochin', 2278, 2178).
28 heuristics('Agra', 'Delhi', 200, 100).
29 heuristics('Agra', 'Hyderabad', 1246, 1146).
30 heuristics('Agra', 'Indore', 591, 491).
31 heuristics('Agra', 'Jaipur', 230, 130).
32 heuristics('Agra', 'Kanpur', 290, 190).
33 heuristics('Agra', 'Lucknow', 369, 269).
34 heuristics('Agra', 'Madras', 2048, 1948).
35 heuristics('Agra', 'Nagpur', 770, 670).
36 heuristics('Agra', 'Nasik', 1005, 905).
37 heuristics('Agra', 'Panjim', 1715, 1615).
```

The status bar at the bottom indicates the cursor is at line 613, column 33, with 4 spaces and UTF-8 encoding. The system tray shows the date and time as 08:18 PM on 01-11-2022.

The heuristics used is **Distance\_between\_cities - 100**.