

## Implementation Details:

1. 2 types of search i.e. DFS and best first search have been implemented.
2. The heuristic used for best first search is the great circle distance: which is the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere (definition src: Wikipedia). Python's geopy library was used to create the facts associated with these heuristics.

## Steps for running the code:

Step 1 : run swi-prolog.swipl edges.pl on the commandline

Step 2: Run initialize.

Step 3: To perform a search type get\_path().

Step 4: Enter the source and destination nodes when prompted (in single quotes), then enter the search type 'DFS' or 'BestFirst'.

```
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- initialize.
true.

?- get_path().
Source Node:
|: 'Bombay'.
Destintion Node:
|: 'Delhi'.
Search Type(DFS,BestFirst) :
|: 'DFS'.
Bombay
Agartala
Ahmedabad
Agra
Bangalore
Allahabad
Bhubaneshwar
Amritsar
Calcutta
Asansol
Chandigarh
Baroda
Cochin
Bhopal
Delhi

distance(23953)
true .
```

We can see that when we do a dfs between Bombay and Delhi we get the above path starting at Bombay and ending at Delhi. The distance of this path is reported at the end in km (23953 in this case).

An example of Best First Search:

```
?- get_path().  
Source Node:  
|: 'Agra'.  
Destintion Node:  
|: 'Meerut'.  
Search Type(DFS,BestFirst) :  
|: 'BestFirst'.  
Agra  
Delhi  
Meerut  
  
distance(266)  
true .
```

The route is Agra -> Delhi -> Meerut and the path length of this route is 266 km.