# Foundations Artificial Intelligence
## Lab - 1
Byreddy Vishnu Saketh

Implementation:

I have taken the average running speed of a healthy man as the maxspeed that a man travels on a paved road. All the other speeds are based on this speed. For the out of bounds condition I used a negligible speed(to avoid any division by zero errors)

A* algorithm considers every pixel to be connected to 8 other pixels around by some edge. Each edge has a cost and each pixel has a heuristic value associated with it.

Each node is a tuple which contains the following values :

Cost + heuristic (Priority Queue is order by this value)

X,y coordinates of the node

shortestCost to that point

TotalDistance to that point

Parent of that node when we take the shortest path

speedMap = {(248, 148, 18): 6.2, (255, 192, 0): 3.1, (255, 255, 255): 4.34, (2, 208, 60): 2.48,
(2, 136, 40): 1.86,
(5, 73, 24): 0.62, (0, 0, 255): 1.24, (71, 51, 3): 8.9, (0, 0, 0): 7.15,
(205, 0, 101): 0.0000000000000000000000001}

**Cost Function** :

My cost function initially calculates the slope and finds the distance based on the slope i.e, distance / cos(thetha). The final cost would be the inclined distance divided by the speed in that particular terrain. The edge costs between each terrain depends on the speed in the terrain. Hence this function takes care of that.

**Heuristic :**

My heuristic function is defined as the distance between the current point and the destination divided by the maxspeed that you can travel in any region. This function makes sure that we are always travelling towards the destination. We are penalized when we try to visit nodes which are far away from our destination.

**Output :**

Output consists of the totalDistance in **meters** of the optimal path given some controlPoints around which we are supposed to travel. Each output has an image associated with it which consists of effects of seasonal changes and also the outline of the path taken colored in red(139, 0, 0).