**Relevant decisions in the implementation:**

The terrain time constants are taken based on the path of terrain. The more difficult the path the greater the value. The terrains which can’t be traveled have the value of infinity. For different seasons the terrain constants change.

**Implementation of cost function**

First I calculated the heuristic distance between the two pixels. Then I added the weights(time constant of terrain) of both pixels and multiplied that by half of distance.

**Implementation of heuristic function**

In this I calculated the 3D euclidean distance taking longitude and latitude distance into consideration so that it is . My heuristic cost is not exceeding the cost of cost function as the terrain time constants are not included in the heuristic function. Hence my heuristic function is consistent.

**Seasonal algorithms:**

**Fall:**

In fall I am checking if there is any paved road or footpath that goes through the easy movement forest. If there is, then I am increasing the terrain constant values of those paved roads and footpaths.

**Winter:**

In winter I am first calculating the edge water pixel and then using BFS to make the 7 pixels within water ice so that it is safe to walk on. Also updating the terrain constant of those pixels to a new value.

**Spring:**

Used similar logic to that of winter season just calculated the pixels within fifteen pixels of water that can be reached from a water pixel with elevation less than 1.

**OUTPUT:**

I am storing the path in a list and for that path I am changing the color pixel and updating that into the image so that the path can be clearly seen.