




25-09-2021

ARTIFICIAL INTELLIGENCE
PRACTICAL 5
ROLL No. 2109805

NAME: JAGRUT GALA
CLASS: TYBSc CS
ROLL No: 2109805
SUBJECT: ARTIFICIAL INTELLIGENCE



Practical 5: Hill Climbing

Q1) Demonstrate Hill Climbing Technique.

Ans:

[p5_hill_climbing.py](#)

```
"""
p5_hill_climbing.py
Author: Jagrut Gala
Date: 14-08-2021
Practical: 5
Objective: Demonstrate Hill Climbing Technique
"""

import math
increment = 0.5
startingPoint = [1, 1]
point1 = [1,7]
point2 = [6,4]
point3 = [5,2]
point4 = [3,1]

def distance(x1, y1, x2, y2):
    dist = math.pow(x2-x1, 2) + math.pow(y2-y1, 2)
    return dist

def sumOfDistances(x1, y1, px1, py1, px2, py2, px3, py3, px4, py4):
    d1 = distance(x1, y1, px1, py1)
    d2 = distance(x1, y1, px2, py2)
    d3 = distance(x1, y1, px3, py3)
    d4 = distance(x1, y1, px4, py4)
    return d1 + d2 + d3 + d4

def newDistance(x1, y1, point1, point2, point3, point4):
    d1 = [x1, y1]
    d1temp = sumOfDistances(x1, y1, point1[0], point1[1], point2[0], point2[1],
point3[0], point3[1], point4[0], point4[1])
    d1.append(d1temp)
    return d1

def newPoints(minimum, d1, d2, d3, d4):
    if d1[2] == minimum:
        return [d1[0], d1[1]]
    elif d2[2] == minimum:
        return [d2[0], d2[1]]
    elif d3[2] == minimum:
        return [d3[0], d3[1]]
```

```
elif d4[2] == minimum:
    return [d4[0], d4[1]]

minDistance = sumOfDistances(
    startingPoint[0], startingPoint[1],
    point1[0], point1[1], point2[0], point2[1],
    point3[0], point3[1], point4[0], point4[1]
)
flag = True
i = 1
while flag:
    d1 = newDistance(startingPoint[0]+increment, startingPoint[1],
        point1, point2, point3, point4)
    d2 = newDistance(startingPoint[0]-increment, startingPoint[1],
        point1, point2, point3, point4)
    d3 = newDistance(startingPoint[0], startingPoint[1]+increment,
        point1, point2, point3, point4)
    d4 = newDistance(startingPoint[0], startingPoint[1]-increment,
        point1, point2, point3, point4)
    print (i, ' ', round(startingPoint[0], 2), round(startingPoint[1], 2))
    minimum = min(d1[2], d2[2], d3[2], d4[2])
    if minimum < minDistance:
        startingPoint = newPoints(minimum, d1, d2, d3, d4)
        minDistance = minimum
    #print i, ' ', round(startingPoint[0], 2), round(startingPoint[1], 2)
    i+=1
else:
    flag = False
```

C:\MyStuff\College Stuff\SEM V\Artificial Intelligence\Practicals\practical_5>py p5_hill_climbing.py

```
1 1 1
2 1.5 1
3 1.5 1.5
4 2.0 1.5
5 2.0 2.0
6 2.5 2.0
7 2.5 2.5
8 3.0 2.5
9 3.0 3.0
10 3.5 3.0
11 3.5 3.5
```

C:\MyStuff\College Stuff\SEM V\Artificial Intelligence\Practicals\practical_5>