|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | T. Y. B. Tech (Computer Engineering) |
| Course: | Advance Algorithm Laboratory |
| Course Code: | DJ19CEL602 |
| Experiment No.: | 05 |

**AIM: Implement KD-Tree.**

**CODE:**

import math

class Node:

    def \_\_init\_\_(self, nums):

        self.nums = nums

        self.level = 0

        self.left = None

        self.right = None

def create\_node(nums):

    return Node(nums)

def traverse\_in\_order(curr):

    if curr is None:

        return

    traverse\_in\_order(curr.left)

    print(f"({', '.join(map(str, curr.nums))}) ", end="")

    traverse\_in\_order(curr.right)

def make\_kd\_tree(seq, depth=0):

    if len(seq) == 0:

        return None

    k = len(seq[0])

    dim = depth % k

    seq.sort(key=lambda x: x[dim])

    mid = len(seq) // 2

    mid\_elem = seq[mid]

    root = create\_node(mid\_elem)

    left\_sub\_arr = seq[:mid]

    right\_sub\_arr = seq[mid+1:]

    root.level = depth

    root.left = make\_kd\_tree(left\_sub\_arr, depth+1)

    root.right = make\_kd\_tree(right\_sub\_arr, depth+1)

    return root

if \_\_name\_\_ == "\_\_main\_\_":

    seq = [[6,2], [7,1], [2,9], [3,6], [4,8], [8,4], [5,3], [1,5], [9,5]]

    root = make\_kd\_tree(seq)

    print("Inorder Traversal: ",end='')

    traverse\_in\_order(root)

**OUTPUT:**

****