

Name: Preena Sunil Jadhav

Sap Id: 60004220127

Batch: C2-2

course: Advance Algorithm

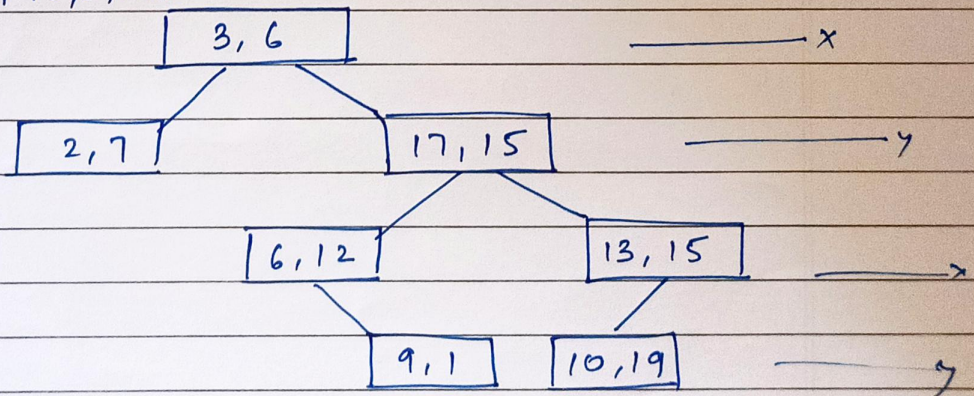
EXP - 5

AIM: Implement KD-Tree

THEORY: A KD-Tree also known as (k-dimensional tree) is a binary search tree where data in each node is a k-dimensional point in space. In short, it is a space partitioning data structure for organizing points in k-dimensional space. A non-leaf node in k-D tree divides the space in 2 parts, called as half-spaces.

Eg:

creation of a 2-D Tree : $(3, 6)$, $(17, 15)$, $(13, 15)$, $(6, 12)$, $(9, 1)$, $(2, 7)$, $(10, 19)$



CONCLUSION: K-D tree have several advantages like efficient search, Dimensionality reduction, etc. It has the time and space complexity of $O(n)$.

Thus we studied and implemented K-D Tree.



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:

```
PS C:\Users\Jadhav\Documents\BTech\Docs\6th Sem\AA\Code> & C:/msys64/mingw64/bin/python.exe "c:/Users/Jadhav/Doc
uments/BTech/Docs/6th Sem/AA/Code/KD_Tree.py"
Inorder Traversal: (1, 5) (3, 6) (4, 8) (2, 9) (5, 3) (6, 2) (7, 1) (8, 4) (9, 5)
```