# kd-tree construction and analysis with OpenMP and OpenMPI

Imola Fodor SM3500474
Foundations of High Performance Computing
University of Trieste

Deadline 28.02.2022

## Contents

**1 Introdution** **2**

**2 Algorithm** **2**

**3 Implementation** **4**

**4 Performance model and scaling** **4**

**5 Conclusion** **4**

# 1 Introdution

A K Dimensional tree (or k-d tree) is a tree data structure that is used to represent points with more than one property in a k-dimensional space. It is a convenient way to organize points by several criteria at once and it provides eg. a convenient way to search, cluster points by their overall similarity.
In this work an effective and efficient *todo add time complexity* way to build and parallelize such a tree is presented.

# 2 Algorithm

todo

**Algorithm 1** Build kD-tree

**Input** arrayOfNodes
**Output** treeRootNode

1: **function** BUILDKDTREE(startNode, length, axis, dim)

2: *myaxis* ← round robin approach between 0 and 1

3: *medianNode* ← MEDIANOFMEDIANS(startNode, startNode + length -1, myaxis, len)

4: *medianNode.left* ← MAKETREE(startNode, medianNode - startNode, myaxis, dim)
5: *medianNode.right* ← MAKETREE(startNode, startNode + length - (medianNode + 1) , myaxis, dim)
6: **return** treeNode
7: **end function**

8: **function** MEDIANOFMEDIANS(startNode, endNode, myaxis, length)
9: **if** $length < 10$ **then**
10: INSERTIONSORT($startNode, length, myaxis$)
11: *median* ← *middleElement*
12: **else**
13: subarrays ← $ceiling(n/5)$
14: allocate array medians of length subarrays
15: **for** $i \leftarrow 1, subarrays$ **do**
16: INSERTIONSORT($startNode, length, myaxis$)
17: *medians[i]* ← *middleElement*
18: **end for**

19: **if** $numSubarrays = high$ **then**
20: *median* ← MEDIANOFMEDIANS($medians, end, myaxis, length$)
21: **end if**
22: **end if**
23: **return** median
24: **end function**

25: **procedure** INSERTIONSORT(startNode, length, axis)
26: similar to the sorting of playing cards in hands
27: **end procedure**

## 3 Implementation

todo

## 4 Performance model and scaling

todo

## 5 Conclusion

todo